Automated Screening of Ultrasound Images for Carcinoma of Liver

by

WUN Yuk Tsan

Systems Engineering and Engineering Management Department

Submitted to the Chinese University of Hong Kong in partial fulfillment of requirement for the degree of

Master of Philosophy

June, 1996

Abstract

Computer-aided clustering is the theme of this thesis which deals with images from ultrasound scans of the liver. The approach is different from previous work. It begins with the consideration of the structure of the target (liver) and the analysis of the characteristics of ultrasonic image of liver. Based on the derived information, statistical features are selected for the task. The conventional statistical parameters used include the mean, entropy and gray difference of the gray levels. Three more statistical parameters are invented for the purpose of the study: the edgeness, roughness and the pixel difference of a subimage. These are second-order statistics which are much simpler to compute, and work much faster, than the traditional cooccurrence matrix.

Observations on the normal liver images lead to the hypothesis that the statistical features (the micro-structures) of the different macro-structures of the liver confirm to a fixed pattern of distribution curves. These curves are stable within a narrow range for all and every image, normal and abnormal. To prove the hypothesis, statistical parameters from normal ultrasonic images are assembled. The distribution curves turn out to be exactly as expected.

The hypothesis leads to the inference that the shapes of the curves enable successful segmentation of the ultrasonic images and satisfactory detection of liver carcinoma.

An image is divided into equal-sized subimages of 10x10 pixels. The different statistical parameters of each subimage are computed and each subimage is assigned

the membership of a particular macro-structure according to the location of parameter along the distribution curve. This works very well with images from the normal liver. The parameters used include the mean gray level, the gray level difference, the entropy of gray levels, the roughness and the pixel difference.

Detection of the tumour continues from segmentation of an image. It takes the advantage of two distribution curves. For the entropy of gray levels, the tumour cells, being most volatile, occupy the high end of the curve. That of the posterior capsule is the next below tumour cells. Determination of the mean entropy of the posterior capsule and taking a threshold from this value will classify a subimage as tumour. To adjust possible mis-classification, the gray level difference curve is utilized. Tumour cells occupy a position in the curve intermediate between the normal cells and the posterior capsule. Another threshold from the mean gray level difference of the posterior capsule determines whether the subimage is a mis-classified normal cell or itself a tumour cell.

Due to the various inherent and inevitable fuzzy nature of the image and the algorithm, there are noises in the result. Warning signals are generated only if the aggregate of suspicious subimages reaches a defined size of 10 which is equivalent to the size of 2.2 cm^2 of the actual liver organ.

The conclusion is: a fast and effective algorithm by analyzing the location of statistical parameters on their distribution curves can segment a fuzzy ultrasound image and detect suspicious tumour sites.

Acknowledgment

I must thank my wife, Vivien, for her tolerance of this post-graduate study. Nowadays, wives have to compete with the colourful faces on monitor screens and the charming sounds from the computer speakers. I would like to express my empathy and sympathy to them but particularly my gratitude to one of them, my own wife.

Before taking this M. Phil course, due to my past experience with biochemical and physio-pathological experiments, I did not understand how experiments could be done in computer science. For this thesis, I have planned and tried several approaches, some fruitless, before reaching the final version. Literature review, logical thinking, and laborious trial by error are of the same nature, whether we are doing biochemical or computer experiments. Thanks to this thesis, I fully understand the frustration, failures and flavour of computer experiments.

My gratitude must go to Professor Vincent Lum for his support of this work and advice on the direction of research. Taking a medical person, without previous formal relevant training, for the M. Phil. in Systems Engineering is itself a courageous experiment. I am also grateful to Dr. Ronald Chung for his constant guidance, and constructive comments in formulating and conducting the experiments. His help is indispensable for this thesis.

I would also thank Mr. S. S. Lam and Mr. C. K. Lam, my fellow-students in the M. Phil. course. I knew nothing about C-language before. Through their help, I peeped a corner of the power of the language; and only through their help, I have done quite a number of programs for the algorithm in this thesis.

Table of Contents

Table of Figures and Tables

CHAPTER 1

INTRODUCTION

1.1 Ultrasonography in Clinical Medicine

The investigation of internal organs by ultrasound is widely used in clinical medicine. It is noninvasive and simple but informative, and is the least expensive of available imaging tests. An inducer delivers a beam of sound wave and receives the reflected wave from the internal organ. The wave pattern is shown on a monitor screen. The ultrasonographer, on identifying an abnormality on the screen, captures the wave pattern onto a hard copy.

The investigative value of ultrasonography depends immensely on the experience and expertise of the ultrasonographer: the direction of the sound beam, the spotting of abnormal features on the screen image, and the persistence of probing a suspicious area. The sensitivity of ultrasonography also depends on the resolution of the sound beam and the size of the target lesion. A small lesion may not give distinct sound reflections or the reflection may not be recognized by the ultrasonographer.

1.1.1 Ultrasonic features of the liver

The ultrasound beam consists of sound vibrations traveling through the tissues. Sound of frequencies greater than 20 *kHz* are called ultrasounds. For diagnostic tests, ultrasound frequencies are in the range of 1-20 *MHz* at the speed of 1.54 $mm/\mu s$.

The ultrasonic image is a cross-section representation of the tissue. The ultrasonography of liver is usually a B- (brightness) mode echography. The strength of an echo is displayed in two-dimension as a variation in brightness along the line of the sound beam. The images produced are two-dimensional cross-sections of the tissues. Highly reflective *echogenic* regions are shown on a cathode-ray oscilloscope as *bright* areas. The major obstacles to the transmission of ultrasound are bones, gas, and excessive fatty tissue; these echo-free structures are shown as dark areas.

The liver is a sonographically complex organ. The liver parenchyma gives a homogenous background within which are fluid-filled tubular structures (hepatic or portal veins), and linear or rounded echogenic areas (fibro-fatty tissue around portal regions consisting of hepatic artery, bile duct and portal vein too small to be imaged). Intrahepatic bile ducts usually do not show up in the image unless they are dilated.

The ultrasonic images are usually displayed in gray scales (new machines now displays colours for the blood flow). They all suffer from coarse resolution and low contrast [Wong93]. The quality of an ultrasonic image depends on extrinsic and intrinsic factors. The extrinsic factors include the resolution and image-capturing ability of the machine, and the brightness and contrast level set by the operator. The intrinsic factors are the macro- and micro-structures of the liver, with different acoustic impedances. Figure 1.1 shows an ultrasonic image of a normal liver in 256 gray levels, 8-bit, 600x800 pixels, and 300 pixels per inch. Figure 1.2 is a sketch drawing of the same image to illustrate the macro-structures of the liver.

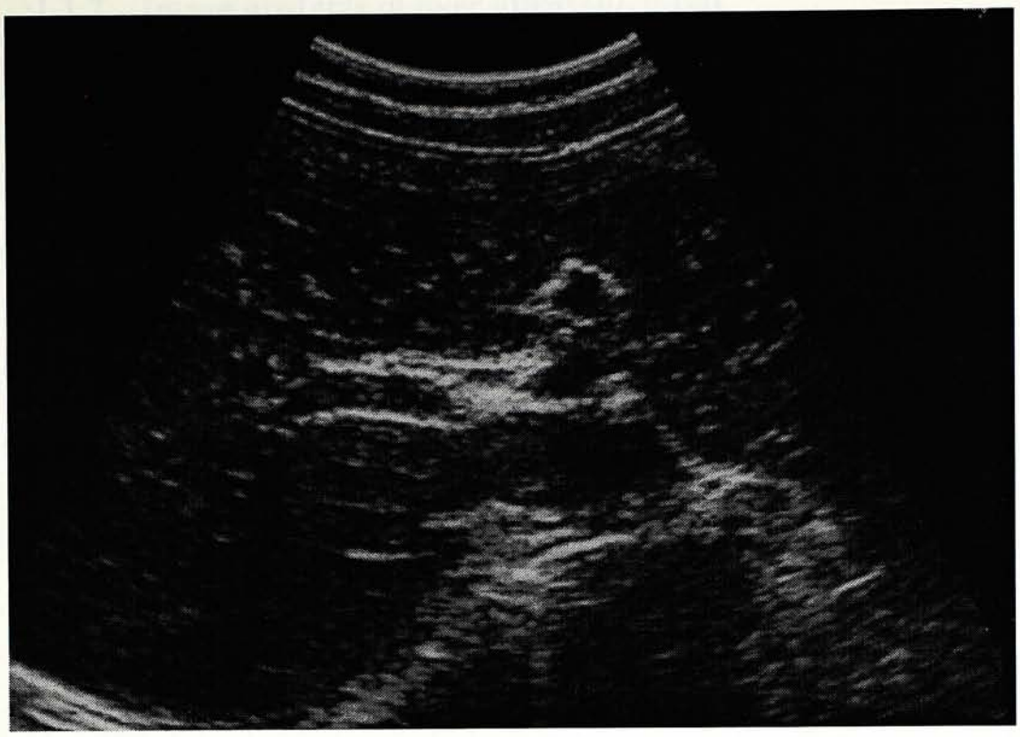


Figure 1.1 An ultrasonic image of the normal liver

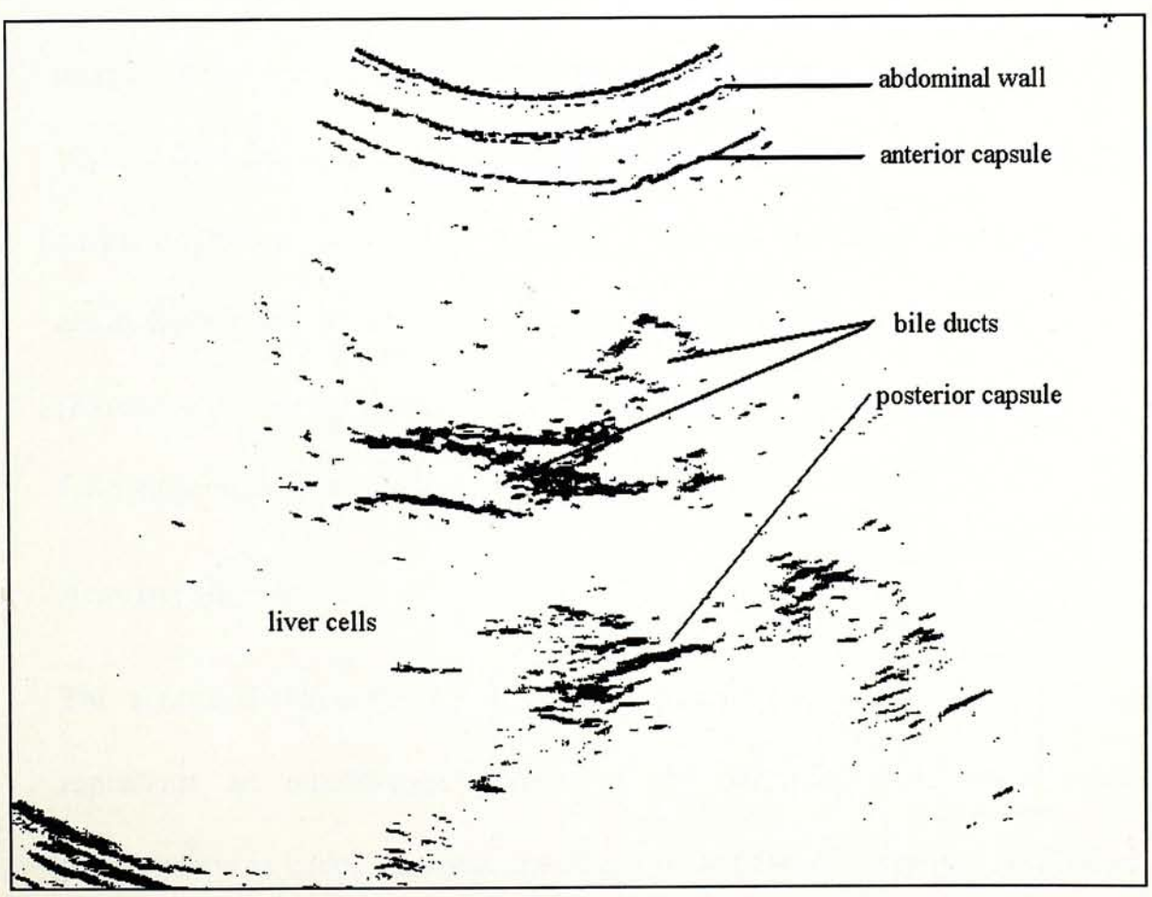


Figure 1.2 Sketch drawing of Figure 1.1

1.1.2 Image artifacts in liver ultrasonograms

When a sound beam meets an interface between one medium and another of different acoustic impedance, the propagation of the wave will be affected. Portion of the incidence beam is reflected while the majority is transmitted or refracted. The intensity of the wave is thus reduced by [Hussey85]:

1. Absorption — ultrasound energy converted to heat along the path of the wave,
2. Scattering — wave reflected away from original propagation path by surfaces of different media,
3. Reflection — some of the energy wave reflected backwards, reducing the downstream intensity.

In imaging, an artifact is anything not properly indicative of the structures imaged. They may cause missing objects (anatomical structures not shown), added objects (images which do not represent real structure), and incorrect object size, shape, brightness, or location. Artifacts may result during the propagation of a sound beam when parts of its energy are absorbed, reflected or scattered. Kremkau et al [Kremkau86] gave detailed accounts of these ultrasonographic artifacts. The followings are a few examples.

Acoustic speckle

The scattered waves may potentiate or cancel one another. The resulting image represents an interference pattern of the scattered distribution scanned. The phenomenon is called acoustic speckle. The apparently excellent resolution close to the transducer is due to acoustic speckle.

Reverberations

If two or more reflecting objects occur in the sound path, multiple reflections (reverberations) will occur. These may be sufficiently strong to cause an image that does not represent a real structure.

Shadowing

When the ultrasound beam encounters a totally reflecting or attenuating structure (e.g. a gallstone), there is no image distal to the obstacle. The absence of image behind an obstacle is referred to as an "acoustic shadow". " ... the eye may become accustomed to the heterogeneity of liver echoes produced by the shadows and thus overlook certain pathologic images" [Weill87].

Enhancement

Enhancement is the opposite of shadowing. It is the increase in reflection amplitude from objects behind a weakly attenuating structure, e.g. an empty gallbladder or a cyst. Both shadowing and enhancement are artifacts useful in determining the nature of the masses. Fluid- or gas-filled masses would give shadowing or enhancement.

These artifacts make segmentation of ultrasound images difficult. For example, edge detection will make the enhancement prominent and disguise the lesion. This feature is illustrated by Figure 1.3 and Figure 1.4 where a liver cyst is missed by edge detection.

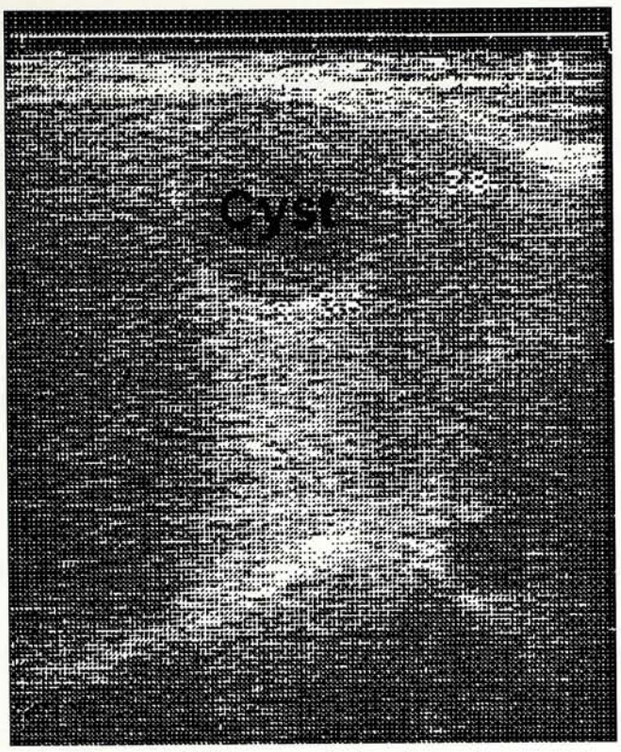


Figure 1.3 Ultrasound showing a liver cyst

Figure 1.4 Edge detection of Figure 1.3

1.1.3 Characteristics of liver ultrasonic image

There is one fundamental but very important difference between ultrasonic medical image and other computerized images like natural scenes. Take the ultrasonic liver image as an example. We have such *a priori* knowledge that we know (or expect) not only *what* will be present (e.g. the capsule, bile ducts, liver cells) but *how* they will present, even with expected modifications called ultrasonic artifacts. Ultrasonographers must learn the normal structure (anatomy) of the organ (e.g. liver) before they can interpret any possible abnormality (pathology). Interpretation of an ultrasonic image by the ultrasonographer is mainly the matching of the image in question against the normal pattern (training set) stored in the human brain. Deviation

from the training set is explained by either matching of stored abnormal pattern or by logical reasoning from previous knowledge (data bank).

Most computer pattern recognition starts with the compilation of training sets or data bank. Of course, the training set may be the expected normal. Parameters which may discriminate the testing image from the expected normal are extracted from either the testing image alone or the testing *and* the training images. The crucial stage is what parameters to use and how to extract them. The success of a pattern recognition algorithm depends essentially on the choice of the distinguishing parameters (or descriptors)[1]. These parameters are usually drawn or derived from a scientific bank of knowledge. There are various methods to help the investigator to include or discard a particular parameter. In the end a large number of parameters may be used or tested though finally a few of them, most often the most complicated ones, are found to be useful. Take the example of analysis of gray scaled ultrasonic images of the liver. We might consider some known features in our scientific bank (the gray level values, the edge detectors, the spatial gray level dependence matrix, etc.) and test them on similar images to select those capable of discriminating the images in question. Figure 1.5 illustrates schematically the procedure. A pattern recognition taking path 1 alone is called unsupervised while that taking path 1 together with path 2 is called supervised.

[1] The terms feature, parameter, descriptor are often used inter-changeably. There are no definitions for them and there is no attempt to distinguish these terms. We do use the term "feature space" to mean the dimension of measurements, but we seldom or even never use such terms as "parameter space" or "descriptor space". I propose to use the hierarchy of feature → parameter → descriptor. A feature is a modality of measurement, e.g. the gray level; the parameter is a member or derived function of the modality, e.g. the mean gray level; and the descriptor is the parameter capable of discriminating objects with the feature. For convenience, these terms are used in this thesis inter-changeably.

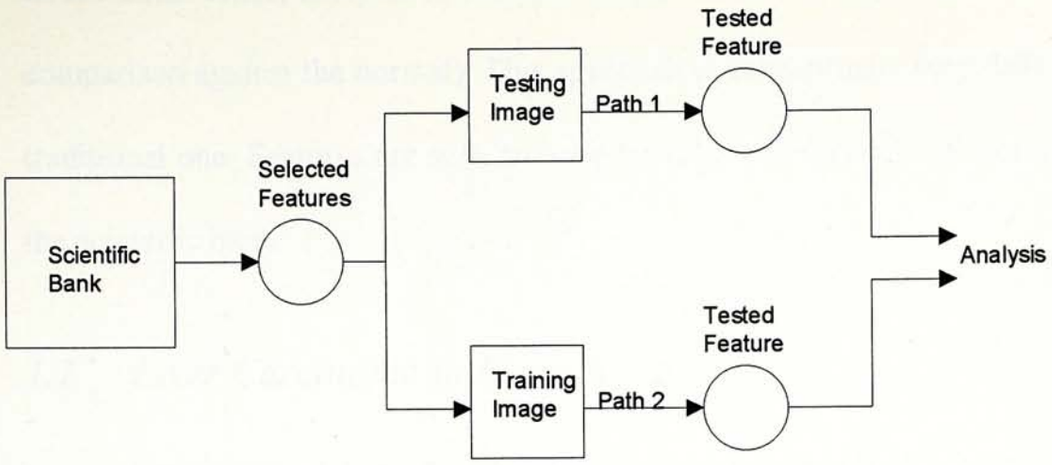


Figure 1.5 Scheme of conventional pattern recognition

Conceptually such an approach is very different from the traditional medical training of understanding the normal (anatomy) thoroughly before analyzing the abnormal (pathology). For medical images, which the normal architecture is known and should be adhered to, pattern recognition should be based on the analysis of the expected normal. Figure 1.6 shows such an approach:

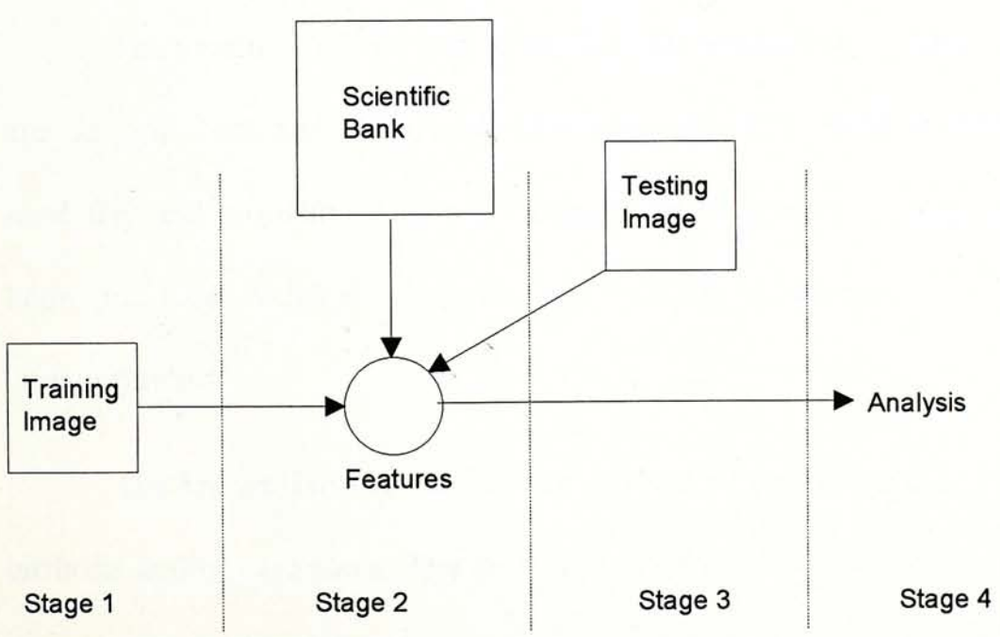


Figure 1.6 Scheme of proposed pattern recognition of medical images

In the strict sense, analysis of medical images should be supervised (once there is a comparison against the normal). This approach is conceptually very different from the traditional one. Features are *selected and tested in reference to the normal* but not to the scientific bank.

1.2 Liver Carcinoma in Hong Kong

Carcinoma of liver or hepatocellular carcinoma (HCC) is the second most common cancer in Hong Kong [HongKong94], next to carcinoma of the group of lung, bronchus and trachea. It usually gives rise to symptoms very late. Chinese are many times more prone to liver carcinoma than the Caucasians, probably due to the high prevalence of carrier state of hepatitis-B surface antigen (HBsAg). In Hong Kong, about 10% of the adult population are HBsAg carriers [Chang85] who have 223 times greater risk of developing liver carcinoma than the rest of the population [Beasley81].

The treatment of liver carcinoma is so far unsatisfactory. The main approaches are surgical resection if possible and chemotherapy. Both methods carry high morbidity and mortality. The best management at present is early detection in the hope of total surgical removal, and primary prevention of hepatitis-B by immunization.

The known HBsAg carriers are advised to have regular screening for liver cirrhosis and/or carcinoma. The best available screening tests are serum alpha-foetal protein (much elevated in liver carcinoma) and ultrasonography of the liver. It is controversial which investigation is better and both tests are often recommended to the carriers [Kang92]. Some studies show a sensitivity of 90% for ultrasonography in

detecting small HCC (<2 cm diameter [Sheu84, Okuda81]) or those HCC associated with liver cirrhosis [Cottone83]. Some [Sheu85b] even observed that ultrasonography was more sensitive than alpha-foetal protein in the early detection of HCC.

Taking 10% of the six-million-plus Hong Kong residents as HBsAg carriers, we need at least 600,000 ultrasonograms of the liver annually, or more than 1500 a day. The work load is obviously overwhelming.

1.2.1 Morphological features of liver carcinoma

The gross morphological features of hepatocellular carcinoma (HCC) form the basis of the appearance of the tumour's ultrasonographic image. Nakashima et al [Nakashima87] classify HCC into four morphological types:-

1. Expansive (Nodular) HCC

   This extends as if it were thrusting intact liver tissues aside. The tumour may be single or multinodular. The characteristic feature is that the tumour is sharply demarcated from the neighbouring cells. Some nodules, especially those arising in liver cirrhosis, are surrounded by a capsule of dense fibrous tissue.

2. Diffuse HCC

   The tumour occurs as numerous small foci, 0.5-1.0 cm in diameter, scattered throughout the liver. They are always associated with liver cirrhosis which itself gives the liver an irregular, multinodular and much distorted architecture, macroscopically and microscopically.

3. Infiltrative HCC

   The typical morphological feature is the irregular and indistinct boundary between the tumour and non-tumour regions. The tumour extends out as if replacing the surrounding cells, in contrast to the expansive type.

4. Mixed infiltrative and expansive HCC

   A primary focus with distinct fibrous capsule spreads out beyond the capsule and infiltrates the surrounding cells.

While the computer image of an expansive tumour gives distinctly different texture and edge from those of surrounding liver cells, the image of an infiltrative tumour may not give a distinct edge. The separation of images of diffuse HCC from that of the surrounding cirrhotic liver may be very difficult.

1.2.2 Ultrasonographic features of liver carcinoma

HCC has no specific sonographic appearance. A small HCC is mostly hypoechoic and very difficult to be differentiated from a haemangioma or metastatic liver cancer. A tumour around 1 cm in diameter is quite difficult to be differentiated from a small liver cyst or a hyperplastic nodule. Solid tumours cause hypoechoic images while hyperechoic lesions are seen in two types of tumour: those with fatty necrosis and those with marked sinusoidal dilatation [Tanaka83]. Sheu et al [Sheu85a] described two ultrasonographic growth patterns for HCC:

1. Major pattern — Small HCCs are hypoechoic. As they grow, they become more echogenic and become hyperechoic when they reach a large size.

2. Minor pattern — The carcinoma is diffusely and homogeneously hyperechoic throughout the course.

The liver is a frequent organ for metastases from various primary sites: breast, bronchus, colorectum and melanoma. There is again no specific feature for these metastases. They may be echo-free, hypoechoic, hyperechoic, hypoechoic with hyperechoic area around, hyperechoic with hypoechoic area around, or other combinations [Rosenbusch89]. Differentiation between HCC and metastatic tumour on an ultrasonic image is usually very difficult.

1.3 Ultrasonography and Computer

The disadvantage of ultrasonography is its low specificity. Ultrasonography is operator dependent, leading to considerable observer variation and difference in diagnostic accuracy [Gosink79, Debongnie81]. Many factors may influence the performance of the ultrasonographer: experience, alertness, pressure of available time, just to mention a few. The human eye may become accustomed to the heterogeneity of liver echoes and thus overlook certain pathological features (acclimatization and fatigue). It is also machine dependent. In ultrasonography the value of pixel depends, apart from the tissue properties, crucially on external parameters such as the settings of ultrasonic scanner. In conventional physician-based diagnosis, the ultrasonic images are optimized with respect to the visual perception of the physician; the parameter settings are frequently changed with different patients, cross sections, and examiners.

The computer systems do not suffer from the drawbacks of acclimatization and fatigue of a human operator. Their measurements are objective and reproducible

[DaPonte89, Thomas91]. "Human observers are good at extracting information relating to contrast and brightness but they are poor at the second-order task of texture analysis; this is better done by a machine that can derive numerical data" [Wells89]. The computer can extract such features for display to help medical decision-making. The human brain on the other hand has capabilities not matched by computers: such as rapid visual recognition, high-speed spatial relational analysis, creativity, three-dimensional visualization, understanding and formulating generic concepts [Ledley88].

Computer technology is widely used in the recently developed medical imaging methods like the computer-assisted-tomographic scans (CAT-scans), magnetic resonance imaging (MRI). But there is very few computer-aided technology in ultrasonography. There are systems for the processing and analysis of data from ultrasonograms [Carrasco86, Ott86]. It is remarkable that some systems are developed on personal computers such as IBM PC, e.g. the DUMAS (Drexel's Unix-based iMage Analysis system [McEachon89], trabecular microanatomy of bone sections [Aaron92], quantitative analysis of biological shape [Zhang92], automatic visual field testing [Filligoi92]. The trend is to make computer vision practicable and affordable to even small laboratories and medical practices. However, there are very few automated computer systems for analyzing the ultrasound images, though they are very essential for massive screening of the general population.

1.4 Objectives of Thesis

In summary, we know that

(1) A high proportion of Hong Kong residents are hepatitis-B carriers who have much greater risk of developing liver carcinoma.

(2) For a possible cure, a carcinoma needs to be detected at an early stage when it is small in size and has not yet disseminated. Hence hepatitis-B carriers should be screened annually or half-yearly for liver carcinoma.

(3) Ultrasonography is the least expensive, most convenient and probably also most sensitive screening test for liver carcinoma.

(4) The effectiveness of ultrasonography largely depends on the ultrasonographer.

Since huge number of patients are at risk, automation of ultrasonography would boost the effectiveness of screening for liver carcinoma in this locality. One of the feasible aspects of automation is the self-analysis of the output ultrasound image by the machine.

The objective of this thesis is to design a system to screen through liver ultrasonograms for abnormalities suggestive of HCC. The ultimate aim of the system is a real-time analysis to signal an alarm when it detects a suspicious region. This will alert the ultrasonographer for further probing of details. Another alternative but less desirable approach is to set the ultrasound machine to take standard series of images automatically. The system then analyses the stored images and reports accordingly. The system is meant to complement and assist, but not to replace, the human expert. "Because of potential misclassification by any automated technique and because of

the physician's responsibility for the diagnosis, the physician should have the possibility of supervising the computerized diagnostic process" [Schuster88].

As the aim is to screen a large population, the system should be (a) fast, (b) inexpensive, (c) easy to be incorporated into existing machines, and (d) sensitive. The trade-off for being sensitive is having relatively more false positives. But it is safer to err on false alarms than to miss the lesion. These are also the aims of the design.

1.4.1 Hypothesis of the thesis

The hypotheses of this thesis are thus: (a) a computer system can, comparable to a human expert, analyze the liver ultrasonograms to detect a small abnormality compatible with HCC, (b) study of the features of images from the normal liver leads to an efficient algorithm of pattern recognition of normal as well as abnormal images. Corollaries of Hypothesis (b) will be further developed in Chapter 3 after more discussions on pattern recognition of ultrasonic images of the liver.

The boundaries of these hypotheses are the aims of the design stated in the preceding paragraph: speed, cost, sensitivity and ease of integration.

1.4.2 Methods of experiment

The ultrasonograms of patients followed at a hospital for a year or more and reported by a radiodiagnostic specialist as normal USGs are randomly collected. All those USGs reported as containing carcinoma of the liver in the same hospital and reported by the same specialist are recruited. All the images are obtained from the same ultrasound machine to offset the machine-dependence variations of USGs. These USGs are scanned with a commercial software into 8-bit, 256-greylevel, 600x800 pixels image files at 300 pixels per inch resolution. They are then digitized

into raw image data containing the gray level of each pixel. These data are normalized with equalized histogram of gray levels to enhance the contrast and to standardize all the images in order to minimize the physical variations due to operator adjustments. Enhancement of the contrast also serves to bring up features otherwise hidden among the gray levels around the maxima of the unequalized histogram [Ballard82]. The conventional histogram equalization transformation is used:

$$y_i = \frac{256}{N} H(x_i)$$

where y_i is the transformed gray level, N is the total number of pixels, and $H(x_i)$ is the cumulative distribution of the original gray level x_i and there are 256 gray levels. (O'Gorman argued that for X-ray, and hence ultrasonic images, the lowest occupied histogram bin is often very large due to the large black background. The conventional transformation method forces the lowest bin away from zero intensity resulting in a lower contrast image [O'Gorman88]. He demonstrated a modified method for optimal intensity range utilization. His calculation, though theoretically sound, has not been widely used nor tested. The thesis uses the conventional method.)

The equalized image is divided into non-overlapping subimages[2] of size 10x10 pixels for processing. The optimal size of the subimage "must be large enough for the estimation of the parameters to be correct and small enough to be able to detect small regions in the image" [Mosquera92]. The choice of the subimage size is difficult as there is very littel similar work on liver USGs in the literature. The present size of 10x10 pixels is finally determined by experimentation and with the consideration of

[2] A subimage is a spatially connected partial group of pixels of an image [Fukada80].

clarity of presentation and reconstruction of the resultant image[3]. A small-sized subimage increases the computer time but also increases the sensitivity of detecting regions of abnormality. With the present resolution of 300 pixels/inch, 10-pixel of the image is equivalent to 0.22 cm of the liver. A single subimage with features suggestive of tumour may be the result of image noise but a region of, say, 10 such neighbouring subimage is very likely to represent actual tumour. The minimal size of such region reflects the sensitivity of the system to detect tumour. Because of the low resolution of the ultrasonic image, noise during image processing is inevitable and a large subimage size include lots of noise effects.

1.5 Organization of this Thesis

This thesis consists of six chapters. The present chapter describes the background information and motivation for the research. There are minimal medical terms and the hypotheses are stated briefly. The full version of the statistical hypothesis will emerge after more detailed discussion on texture analysis.

Chapter 2 reviews the past and recent computer analyses on ultrasonic images. Though computer vision in medical images is abound, studies on liver ultrasonogram are limited in quantity, especially on automatic textural analysis.

I shall discuss statistical texture in Chapter 3. Preliminary observation and analysis of some statistical features lead to the full hypothesis of this thesis. I shall evaluate some traditional statistical features according to the objective of this thesis, and a few new features to meet the objective.

---

[3] Sizes of 25x25, 20x20, and 10x10 have been experimented. The shapes of parameter curves (Figure 3.3) are the same with different sizes of the subimage.

In Chapter 4 the hypothesis is tested on ultrasonic images of a normal liver. As a result, an algorithm is developed to segment the ultrasonic images. Successful segmentation supports the hypothesis and finally leads to the objective of the thesis in the development of an algorithm for the detection of suspicious tumour in such images.

Chapter 5 describes the algorithm for computer assisted detection of liver tumour in ultrasonic images. The algorithm is to be tested on abnormal as well as normal images for its effectiveness.

Finally Chapter 6 summaries and evaluates the findings. It also discusses the possibility for future research and development.

CHAPTER 2

COMPUTERIZED MEDICAL IMAGING:

A REVIEW

2.1 Computer Vision and Medical Imaging

Many medical investigations, e.g. biopsy sections, X-rays, echograms, must be processed by repetitive human visual examinations. The advent of the first computerized tomographic scanners in 1973-1974 marked the application of computer vision[4] in clinical medicine especially radiodiagnosis, radiotherapy and pathology. "Perhaps principally due to the spectacular advances made in computer and image processing technology, there has been a virtual explosion in applications of imaging methods to medicine" [Ledley88].

There are seven fields in computerized medical imaging [Ledley88]:-

1. digital image engineering
2. image acquisition methods
3. storage and transmission of images
4. image processing and manipulation
5. image display methods
6. pattern recognition
7. real intelligence and image interpretation.

[4] "Computer vision is about image acquisition, processing, classification, recognition, and, to be all embracing, decision making subsequent to recognition" [Low91].

This thesis deals with image interpretation (detection of liver tumour) through pattern recognition (comparison against the training set of normal liver images) after image processing (extraction of discriminating features). In practice, these fields are the different phases of one piece of work — image analysis.

Pattern recognition

Pattern recognition refers to the computer analysis of images for object recognition. There are three phases.

(a) Feature Extraction. The main features of the objects to be recognized are assembled. *Filtering* reduces unwanted spatial frequencies (noises or artifacts). *Edge enhancement* brings out the boundaries between different objects or structures of interest from surrounding tissues (which are suppressed). *Texture analysis* involves neighborhood operation when the gray (or colour) value of each pixel in the original image is replaced by a value computed from neighbouring pixels (convolution).

(b) Context Processing. The objects are measured to obtain quantitative value, or are compared with previously stored patterns.

(c) Pattern Classification. The objects are classified according to pre-determined categories. This process is also called segmentation which is the process of labeling pixels as members of different regions.

There are two broad categories of methodology in pattern recognition [Coleman79]. The first one is knowledge-based, closely related to computer artificial intelligence. The second is mathematics-based using statistical analysis.

2.1.1 Artificial intelligence

Artificial intelligence uses language theory to describe the primitive elements of objects and their relationship. Such systems, an expert system being one, have two advantages: (a) the rules (languages) make full use of the past experience of a human expert, (b) computer-generated check-list ensures comprehensive search. Expert systems have been used in several aspects:

1. An interface between computer vision and human expert
2. A tool in choosing the suitable methods for processing or analysis
3. An assistance in decision-making, incorporating the results of computer vision and the human knowledge in specific domains,
4. A mixture of the above.

For illustration, the following is a brief description of examples (though not necessarily representatives) in each aspect.

Interface between computer and human expert

Expert systems recommend to the user (a medical scientist) the appropriate, among the numerous available, algorithms at different stages of image processing. For example, in the Labo Image [Hu90] the user expresses his goal and the system suggests the segmentation method with explanation if required. An interactive menu guides further processing, step by step, till the user is satisfied with the results meeting the initial goal.

Expert system to guide image processing

The system by Chen et al [ChenLin89] translates brain images from X-ray computerized tomogram (CT), magnetic resonance imaging (MRI), and positron emission tomogram (PET) into semantically meaningful entities, together with knowledge about the characteristics of these investigative scanners, the anatomy of the human brain, brain diseases, and the different methods of imaging analysis. The system chooses the segmentation algorithm with the highest certainty factor to show an anatomical structure of interest.

Object-oriented Image Expert System Shell (OOI)

The system first segments a digitized image and then interprets the extracted structure, using image object models pre-defined by users. The description of the segmented regions, the relationship between each region, and the information about the analyzed image were physically linked in an interpretation lattice. The lattice matches all these data with the user's query to select the conclusion having the highest certainty factor. An example of such a system is the medical expert system for cervix uteri morphometry for the diagnosis of cancer cells in cervix uteri [Leung93].

Integrated knowledge bases

Tjahjadi et al [Tjahjadi89] devises a system consisting of three knowledge bases for pattern recognition, description builder and knowledge retention/retrieval. It selects the appropriate algorithms (edge detection, statistical correlation, symmetrical analysis and function) for processing basing on the complexity of an image. The description builder matches the functional description of the image against the stored

facts in the knowledge retention base to generate a preferred hypothesis of pattern recognition.

2.1.2 Mathematics models

Mathematical models build from numerical data consequent to digitalization of an image. Each picture element (pixel) is represented by a datum (gray level) or a set of data (red-blue-green colour). Many numerical data can be further generated relative to the neighborhood (convolution) or as a member of an area (subimage). The number (dimension) of such feature spaces can be numerous. The difference in the numerical feature space is not directly proportional to the human perceptual difference. In other words, the mathematical model may detect differences imperceptible by human beings.

Image filtering, edge-enhancement and texture analysis described above are examples of mathematical approaches. Examination of the artificial intelligence models will find that most of them have incorporated mathematic information in their systems as the initial steps in image description. Under the mathematics models, some researches use segmentation algorithms while others use neural networks.

Neural Networks

Neural networks are a relatively new computing discipline but have been widely used in pattern recognition like automated character recognition. The system is capable of adaptive response to information environment through inter-connected elements (neurons and nodes). On the basis of experience (training), the system organizes itself to enable extrapolation when faced with new, yet similar, pattern. It transforms the input information to the output inference.

Though the advantages and disadvantages are not well understood, they seem to be particularly well suited for pattern associated applications [Schalkoff92]. Different kinds of network have been tested for ultrasonic images [Conrath89, Schouten93], and there are good results on liver USGs [Zatari93]. Large amounts of training data are usually necessary and sometimes phantom data are used [Botros92].

The neural network is not suitable for the proof of the statistical hypotheses of this thesis. The hypotheses (Chapter 3) need theoretical as well as implementational proof. The neural network is "a non-algorithmic, black box strategy" [Schalkoff92]. There is no answer yet to the question of how a neural network does its job [Hecht-Nielsen89]. A network can prove that the hypotheses do work but cannot show how they work.

Segmentation algorithms

There are two approaches to segmentation: (a) differentiation and (b) clustering[5]. Differentiation finds the separation boundaries (edges) between objects (regions), and is inevitably sensitive to noises. We have seen that edge detection may mislead the identification of objects in Figs. 1.3 and 1.4. This will be further discussed in Chapter 3. Clustering groups a given set of objects into subsets of similar properties (these subsets are called clusters). *Regions of the image that appear the same are expected to have similar properties (features) which should enable clustering.*[6]

[5] Segmentation is labelling *pixels* as members of different regions. Clustering is separating different groups. But Bow defines clustering as "the nonsupervised classification of objects. It is the process of generating classes without any *a priori* knowledge of prototype classification" [Bow92]. In general, clustering is not constrained to supervised or non-supervised classification. This thesis uses the general meaning for this term.

[6] This forms the basis of the second hypothesis of this thesis. Chapter 3 will elaborate it further.

2.2 *Computer Vision and Ultrasonic Images of Liver*

Computer analysis of liver ultrasonogram (USG) has so far been using two kinds of data:-

1. radiofrequency data (one-dimensional A-mode scan)
2. amplitude derived data (two-dimensional B-mode scan)

There are however only a limited number of researches on pattern recognition of liver USGs and some of them will be reviewed.

2.2.1 Studies on radiofrequency (RF)

Lerski et al [Lerski79] used the echo amplitude and derive five texture parameters from the ratio of number of peaks present in unsmoothed signal to the number present after smoothing of a preset power. They assembled these features from normal controls and from confirmed abnormal livers (alcohol induced diffuse liver diseases). The data were manually taken with the ultrasound beam at chosen directions, avoiding major vessels, bile ducts, the beginning and the end of the reflected beam (the capsules). The minimum distance (Euclidean) classifier was used to differentiate the abnormal from the normal. It should be noted that their hardware system consisted of 32K computer memory for processing and 20 *MHz* 8-bit processor for digitalization.

Itoh et al [Itoh85] digitized the RF intensity (ranged from 1 to 63 db) to construct histograms from selected areas of interest (ROI) in liver USGs. They expressed the histogram width in terms of decibel at 0%, 10% and 50% of normalized frequencies (the maximum frequency of the echo intensity for each ROI adjusted to

be 100%). They found that the histograms of liver carcinoma were more widely dispersed than those of normal liver.

The acoustic coefficient and the backscattering coefficient are functions of RF. Botros compared these functions at different depths of the ultrasound beam of testing images against references taken from normal liver scans, using again Euclidean distances [Botros88]. His algorithm was implemented on a microcomputer connected to an ultrasound scanner and successfully tried on 15 subjects with different types of liver abnormalities.

Momenan et al took ROIs from USGs and divided each ROI into several subregions from which they computed four parameters: average distance between regularly positioned specular scatters, ratio between specular to diffuse backscatter intensities, variability in specular backscatter intensities, and the slope of attenuation coefficient [Momenan88,89]. Similar parameters were obtained from reference tissue types of normal controls. The distance from the reference was used for classification of abnormality, supplemented by T^2 hypothesis test to detect heterogeneity in tumours.

2.2.2 Studies on amplitude derived data

Raeth et al [Raeth85] used 2D texture information to analyze liver USGs (from normal controls, and patients with diffuse parenchymal or malignant liver diseases). They took no more than one ROI of 3.7x3.7 cm from each USG and computed a large set of first- and second-order statistics. From these they calculated the probabilities and distribution densities from 20 healthy human controls. They classified testing images against the training set with Bayesian decision procedures

and achieved an overall accuracy of 98% and 89% in classifying diffuse and malignant liver disease respectively.

Schuster et al [Schuster88] employed local texture analysis for each pixel; more than 35 parameters were calculated from five categories: textural edgeness, gray-level run length, matrix of co-occurrence, spread, and relative extreme density. Whenever a suspicious area was encountered, a reference image of a proven diagnosis was copied from diskette to the monitor screen. The suspicious ROI was classified as "same texture" or "different texture" as the reference pattern.

Personal computer was used by DaPonte et al to grab images from the ultrasound scanner [DaPonte89]. They then isolated ROIs of 64x64 pixels close to the centre of the image, again avoiding blood vessels to compute 23 statistical parameters. T-test and Mann-Whitney tests successfully classified 15 images out of 16 (10 normal and 6 abnormal). Youssef et al also employed personal computers to digitize the USGs into 512x512 8-bit buffer storage [Youssef89]. They selected ROIs of at least 1.5 cm^2 for computation of statistical parameters. Images from carcinoma of liver showed low average contrast of the cooccurance matrix. Kim et al [Kim91] also selected ROIs from 512x512x8 bits digitized images. But they defined a new statistical feature: the run difference matrix which was an extended and generalized version of gray level difference. They also used personal computers for their algorithm although they employed neural networks for the analysis.

A new mathematical approach was employed by Akiyama et al [Akiyama90]. Their ROIs were areas of 64x64 pixels (1.7x1.7 cm^2), four such ROIs in each image. The average fractal dimension from these ROIs was highest in images from fatty livers, rather than from cirrhotic or normal livers.

The final example is from the unique work by Layer et al [Layer90] who tested the fat distribution in liver by chemical analysis of histological sections against ultrasound image analyzed with statistical parameters. They took irregularly-shaped ROIs of 800 to 1200 pixels, one from each scanned image (256x256 8-bit), again avoiding the large blood vessels. They found that the mean grey level describing the image brightness was the most powerful parameter in differentiating normal and the nodular fatty liver found in liver cirrhosis.

2.3 *Implications of Previous Work*

That there are so many different approaches for pattern recognition means there is no "the best method" for the problem at hand. No matter which approach taken, the researchers usually start with statistical transformation which seems to be *the* fundamental step. Artificial intelligence is theoretically attractive in that it incorporates and interacts with human experience and expertise. It also builds from the statistical processing of images. Rules (language) are put on top of mathematics for decision-making. If mathematical approach can solve the problem at hand, the system will work faster — a requirement for mass screening. On this reasoning, this thesis chooses to take the mathematics instead of the language approach. Indeed, most researchers adopt this approach and obtain satisfactory results for their purpose.

The choice between neural network and mathematical segmentation is difficult as supporter of either approach can cite lots of experiments with satisfactory results. Considering the boundaries of the objectives and hypotheses of this thesis, a mathematics model using clustering approach is most suitable, and textural analysis rather than edge detection is used. If the hypothesis elaborated in Chapter 3 is not to be rejected, a neural network is not necessary even for implementation of the system.

The obvious drawback of the mathematics approach is the n-dimension of the possible feature spaces. Most researches use a large number (n>10) of statistical features most of which are large-sized matrices which burdens computation. The fact that so many different features and parameters have been tried reflects the truth that the choice of feature spaces is domain-dependent and traditional features may not be adequate. Apparently the mathematics approach may need complicated computation and algorithm. Yet this thesis is to show in the end that simple algorithm and feature spaces may be adequate for satisfactory pattern recognition of liver carcinoma.

Some researchers [Lerski79, Botros88, DaPonte89, Youssef89, Kim91] used simple hardwares like a personal computer in their system, for image acquisition, digitalization and even processing. Their aims were to incorporate their systems into existing ultrasound scanners. This does not only facilitate the ultimate acceptance by the medical profession to use computer systems in ultrasonography but also encourages them to experiment with computer vision during the process of developing the final version of the computer systems. Medical personnels are usually not familiar with the mathematical and computational theories; any apparatus apparently more sophisticated than a personal computer may deter them from taking active parts in experiments and testing of the system.

In short the objectives of this thesis are not only probable but also pragmatic.

2.4 *Limitations of Previous Work*

As discussed above, the feature spaces are potentially large and there is no guide as what feature spaces to choose. What have been achieved are the methods to test the validity of the feature spaces (c.f. Section 3.2 Chapter 3). There is no scientific method to guide what features to choose and how to choose them. Researches still resort to trial by error for the suitable features to solve the problem at hand.

We can see from the methodology of the researches described above that features are drawn from the known scientific bank (or rarely new features devised). These features are then extracted from the testing images with or without normal ones. This fits into the "conventional model" represented in Figure 1.5. Because of this model, large dimension of feature spaces (most of which may be redundant) is often tested before implementation. This thesis attempts to use the proposed model in Figure 1.6 to limit the feature space and construct simple algorithms for pattern recognition in ultrasonic medical images of individual organs.

Many of the afore-mentioned researches used ultrasonic images without transformation or normalization though some used histogram equalization. Without standardization of the images (which are operator- and machine-dependent), it is very difficult to extrapolate findings to other settings. Moreover, transformation of any kind alters the statistical properties of features. Chapter 4 (Section 4.1) will demonstrate how gray level histogram equalization changes the standard deviation of the grey levels in a subimage (or equivalently regions of interest).

With the exception of the work by Botros [Botros88] (who had not mentioned explicitly), all the researches selected regions of interest (ROIs) for their algorithm. All non-cell structures were avoided. These researches studied liver cells proper (the "parenchyma" in medical jargon) instead of the liver image. There are consequently two limitations. First, human interaction starts right from the beginning and this makes mass screening of the normal population not feasible. Second, computer vision in this respect is constricted and full automation is not possible. The whole image of the organ should be included in the algorithm and human expert interaction should be left towards the last phase of the screening system. This thesis works towards this aim. In this aspect, the thesis is unique in that there is no preceding or comparable technique.

CHAPTER 3

STATISTICAL TEXTURE

3.1 Statistical Textural Analysis

There is no unique or universally agreed definition of image texture. Loosely, texture is a property of surfaces: smoothness, coarseness, and regularity. Texture can be described statistically, structurally or spectrally. Statistical texture computes the statistics derived from the distribution of local features. A local feature may be, for example, the gray level intensity from which the mean and standard deviation can be derived for any pixel relative to its neighborhood. Structural texture analyses the texture elements (or shapes) of the image: area, perimeter, size, circularity, etc. Spectral texture measures the power distribution in the Fourier Transform (auto-correlation) for periodicity. Generally speaking, statistical analysis is good for fine micro-structures and structural analysis for coarse macro-structures [Tomita90].

The statistics derived from a sample as a whole are called first-order, e.g. the mean, the standard deviation of the gray levels, as they describe local features not related to context. The statistics for a member of the sample in spatial relationship to the surrounding members (a pair of members at a time) are called second-order, e.g. the co-occurrence (spatial gray level dependence) matrix, the gray level difference [Levine85]. The first-order statistics are embedded in the second-order statistics [Chen79] which therefore give more information than the first-order ones but require disproportionally more computational complexity. When the spatial relationship

involves more than one pair of pixels, the statistics are called higher-order statistics, e.g. the run length. Higher-order statistics do not necessarily give better results in texture discrimination than the lower-order ones [Tomita90], and human observers cannot differentiate images with different second-order statistics but identical first- and second-order statistics [Julesz75].

The simplest and also the most basic feature is the image intensity level — the actual gray level of the image pixel. For a region of an image or the whole image, the probability histogram is often used in image processing for its ease of modification and for comparison of different images. The histogram, $h(g)$, is the number of occurrence of gray levels in the image of N pixels. Let $\phi(g)$ $(g=0, \ldots, 255)$ be the number of points whose intensity is g (ranging from 0-255 in the 8-bit gray scale) in the image, then

$$h(g) = \phi(g) / N$$

Let $\delta=(r, \theta)$ be a vector in the polar coordinates of an image such that two pixels are separated by a distance r and at an angle θ. The co-occurrence matrix of the image is the probability, $P_\delta(i,j)$, of the pair of gray levels (i,j) at pixels $I(i)$ and $I(j)$ occurring at separation of δ [Haralick73]. For a 256-greyleveled image, the difference $(i-j)$ ranges from 0 to 255. The probability $P_\delta(g)$ $(g=0,\ldots,255)$ for an image of size $m \times n$ pixels is

$$P_\delta(g) = \sum_{g=0}^{255} \sum_{i=o}^{n-1} \sum_{j=0}^{m-1} I_g(i) - I_g(j)$$

A run is a set of vertically, horizontally or diagonally neighbouring pixels displaying similar gray levels. The run length matrix $R_\theta(l, g)$ (l=1,...,N, g=0,...,255) is the frequency that l pixels with gray level g continue in the direction θ=[0°,360°], N being the total number of pixels in the image [Galloway75, Levine85]. The run lengths reveal both directionality and coarseness of image textures and gray level runs can be computed very easily. It is sometimes called third-order statistics [Tomita90] as it calculates the spatial relationship of more than two pixels at a time.

Many different statistical descriptors have be derived from these three basic features. They, either alone or in combination, segment successfully a wide range of images.

3.2 Statistical Texture for Segmentation

This research uses the statistical approach because the features are in most cases simple to compute, and because statistical pattern has been widely shown to be effective in segmentation. To cite a few examples, Lerski et at [Lerski79] studied alcohol induced diffuse liver disease; their algorithm achieved an accuracy of 95% in diagnosis but human experts achieved 73% at the best. Mailloux et al [Mailloux83] had 87% accuracy in differentiating normal from diffuse liver diseases, and Lizzi et al [Lizzi83] had similar accuracy in distinguishing cirrhotic and malignant liver tissue. Raeth et al [Raeth85] showed that statistical pattern recognition was better than conventional real-time scanning. By selecting 40 specimens randomly, they tested the accuracy of computerized analysis against the performance of three experienced human experts. The image analysis achieved accuracy of 95% and 90% in diagnosing diffuse disease and tumour respectively while the human experts achieved 85% and

97% respectively. All the human experts failed in diagnosing chronic hepatitis. Insana et al [Insana87] showed that second-order statistical properties discriminated the normal liver from the chronic active hepatitis. Schuster et al [Schuster88] classified normal and fatty liver with accuracy above 90%.

While there is no doubt about the ability of statistical patterns in segmentation, there are two difficulties, one common to statistical texture analysis, and the other specific to ultrasound images. Due to the low resolution of ultrasonograms and the biological variability in human tissue, numerical values of tissue texture will largely overlap. "A single texture property is hardly sufficient for tissue differentiation and consequently a series of parameters have to be calculated" [Schuster88]. On the other hand, the success of segmentation is strongly dependent on the choice of statistical features [Momenen88]. It may be difficult to determine the feature space which characterize texture so that segmentation algorithm can automatically segment images using the appropriate set of features. Experimentation (trial by error), stepwise statistical discriminating analysis [Layer90], the number of within-class outliers and among-class overlap produced by each feature [Muzzolini91], direction of the co-occurrence matrix [DaPonte88], and the ratio of between- to within-cluster scatter [Coleman79] have been prosposed. For this research the choice of the feature space depends on the shape of the distribution curve among the macro-structures.

3.3 Statistical Features Studied in This Research

3.3.1 First-order statistics

As this research aims at automated screening of a massive sample, speed is very essential and computational complexity is avoided. Three first-order statistics are

evaluated. They are the mean, standard deviation and entropy of the gray level distribution, computed from the gray level probability histogram *h(g)*:

Mean (μ) $$\mu = \frac{1}{N}\sum_{g=0}^{255} g * h(g)$$

Standard Deviation $$\sum_{g=0}^{255} (g-\mu)^2 * h(g)$$

Entropy $$-\sum_{g=0}^{255} h(g) * \log(h(g))$$

The entropy measures the homogeneity of gray levels; when concentrated at a narrow range of gray levels, entropy will be low. In an 8-bit gray-scale image (such as those studied in this thesis), there are 2^8=256 gray levels. The maximum value which the entropy can attain is 8. Entropy of the gray levels could be defined as the number of bits per pixel needed to represent the image without information loss [Rhys90].

3.3.2 Second-order statistics

Since the second-order statistics are very powerful descriptors, they must be considered in the algorithm.

Co-occurrence matrix

Conners et al [Conners80], comparing the co-occurrence matrix, run length matrix, gray level difference matrix and the power spectral matrix, found that the co-occurrence matrix is the most powerful in automatic texture discrimination. However, the amount of data associated with co-occurrence matrices can rapidly become unmanageable [DaPonte88].

Haralick et al [Haralick73] studied several statistics derived from the co-occurrence matrix in detail. They observed that the angular second-moment, entropy, sum entropy, difference entropy, correlation and maxi-correlation have the most invariant property. Raeth et al [Raeth85] found that the contrast from co-occurrence matrix was efficient for recognition of liver tumours. The angular second-moment, as used by Youssef et al [Youssef89] for liver ultrasounds, is

$$\sum_i \sum_j \{p(i,j)\}^2$$

where $p(i,j)$ is the $(i,j)^{th}$ entry in the normalized co-occurrence matrix. This descriptor is, among the several derivatives of the co-occurrence matrix, less complicated to compute. It was tried on abnormal as well as normal liver ultrasonograms. While the computation takes up a relatively lengthy time and requires RAM (random access memory) outside the scope of a personal computer, it does not give more information than the other second-order statistical features.

Figure 3.1 shows the boxplot of the reciprocals of the angular second-moment of the co-occurrence matrix of tumour cells and normal liver cells. (The angular second-moment detects uniformity. A completely uniform surface gives the maximum value of 1. In order to show the irregularity of structures, I take the reciprocals. Figure 3.1 presents the integer values after being scaled down by 10. The higher the value, the more irregular the subimage is.) It is obvious that there is no significant difference between the mean values of the two cell groups, and that a large area of overlap exists between the two macro-structures.

The lengthy computation (relative to other second-order statistics) and the non-discriminating ability of the co-occurrence matrix makes it unsuitable for the purpose of this thesis.

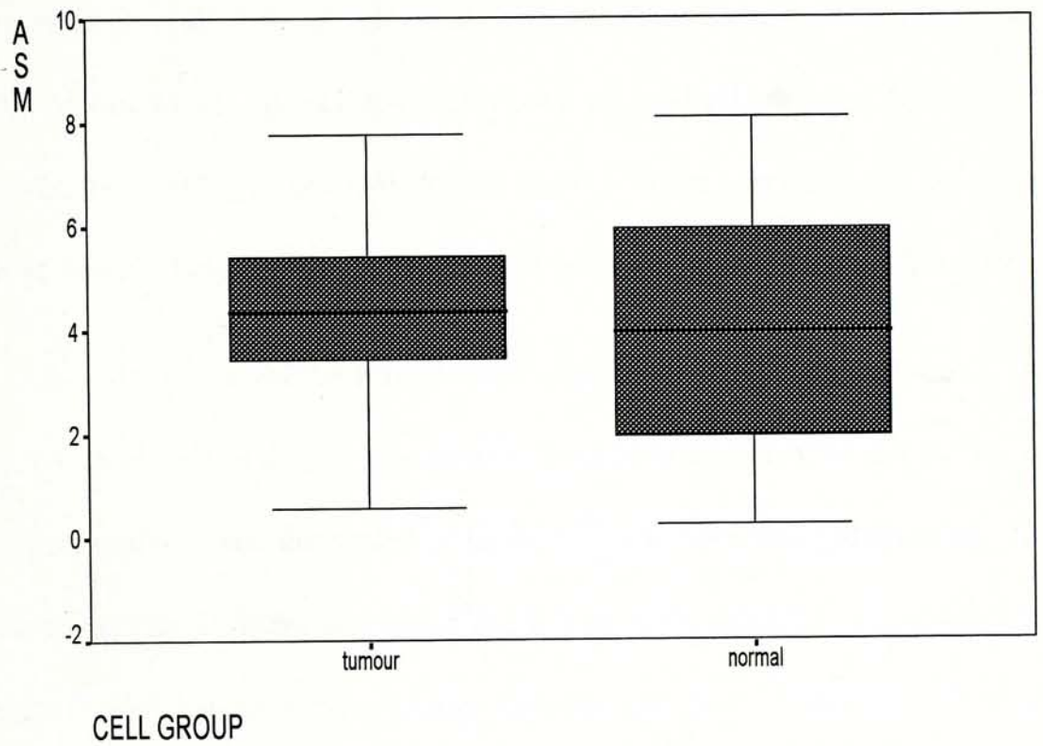


Figure 3.1 Boxplot of the reciprocals of angular second-moment (ASM) of normal and tumour cells from the training set.

Fractal Dimension

Mandelbrot in 1982 first introduced the concept of fractal dimension to estimate the length of coastline [Mandelbrot82]. It is a measure of randomness, describing the ratio of the number of features at one scale to the number of features at the next scale. So it is a second-order statistical technique [Chen89]. The technique has been widely used in texture analysis and segmentation [Pentland84, Keller89, Rao90, Mosquera92]. It has also been used in medical images [Zhang89, Jang90, Fortin92a, Fortin92b, Chan93, Wong93, Ng93]. Some workers used fractals in liver USGs [Chen89, Akiyama90, Albregtsen92] with fair results after modification.

A real surface will be fractal over some aspects of scales rather than over all scales. Medical images do not give perfect fractal surfaces and do not have a constant fractal dimension over all scales [Chen89]. Some workers [Medioni84, Keller89] reported that fractal dimension alone did not provide sufficient discriminating power to classify natural textures. Using fractals on liver USGs Albregtsen et al [Albregtsen92] concluded that fractal dimension was not sufficient for a reliable discrimination between normal liver cells and cells of induced cancer. Some modifications are often used, e.g. the gray level run length at two different quantization widths [Albregtsen92], similarity measurement [Chen89].

From the results by Akiyama et al [Akiyama90], one can observe that, while the average fractal dimensions of different liver pathologies are distinct, there are fair amounts of overlapping values among classes (normal liver 2.359±0.031, fatty liver 2.399±0.031, cirrhotic liver 2.351±0.035). The same phenomenon was observed in this research (normal liver cells 2.66±0.15, tumour cells 2.68±0.13, Figure 3.2). When

tested on clustering algorithms, the fractal dimension[7] was found to have inadequate discriminating power in the differentiating different macro-structures of the liver images.

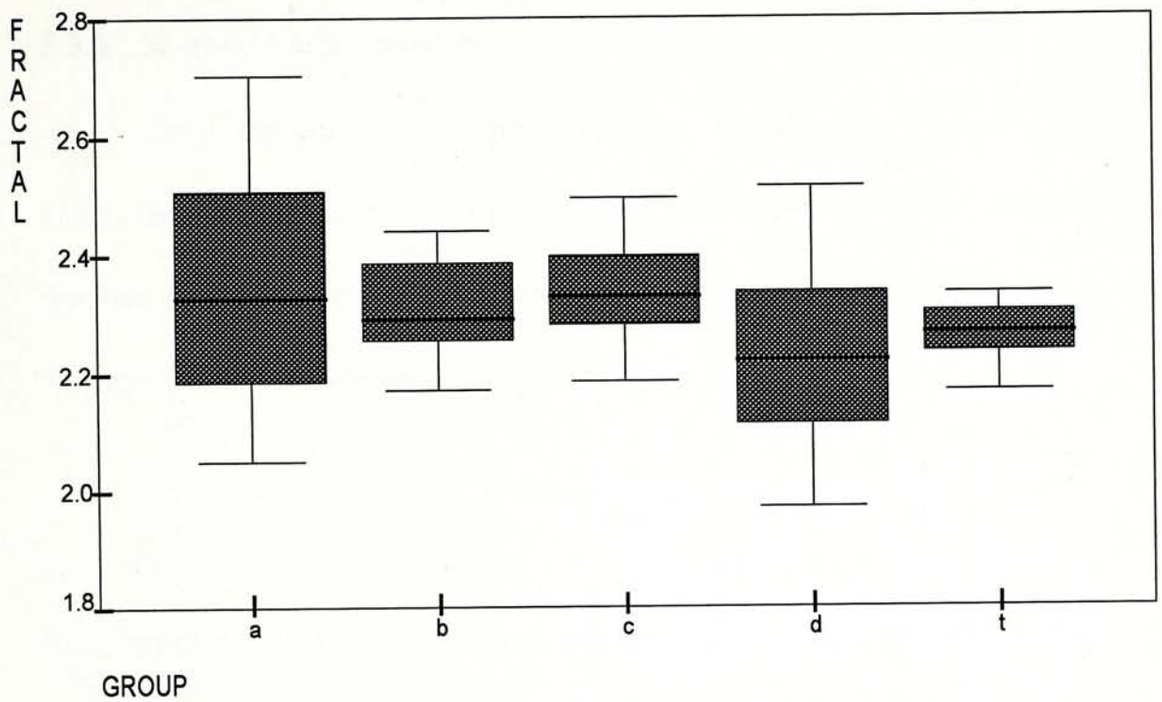


Figure 3.2 Fractal dimensions of different structures from an abnormal liver ultrasonogram. (a=anterior capsule, b=background, c=cell, d=diaphragmatic capsule, t=tumour)

Gray level difference

Gray level difference [Levine85, Weszka76a, Weszka76b] is chosen because of its computational simplicity. Let $P_\delta(g)$ be the histogram of the gray level differences of two pixels separated by the distance δ, then the mean gray level difference is

$$\sum_{g=0}^{255} g * P_\delta(g)$$

[7] There are different methods to calculate the fractal dimension. This thesis uses the differential box-counting method by Chaudhuri et al [Chaudhuri92] for its efficiency

A small mean gray level difference indicates coarse texture having a grain size equal to or larger than the magnitude of displacement δ [Haralick92]. For this research, δ is taken to be 90° (vertical direction along the direction of the ultrasound beam) and 1 pixel apart. (In most cases texture can be discriminated best when r=1 [Chen79].)

3.3.3 Higher-order statistics

Only the run length percentage is tested. The run length percentage characterizes the distribution of runs. In a homogeneous image, when only a small number of runs is present, this feature is small. With reference to the definition in Section 3.1, the run length percentage is

$$\frac{1}{N} * \sum R_\theta(l, g)$$

This parameter is fast to compute and has been used with some success by other investigators [Raeth85, Layer90]. This research has tested it but finds it unstable among different ultrasound images. Layer et al [Layer90] also found the run length less useful than the mean gray level. The final algorithm excludes it as a discriminating feature (Chapters 4, 5).

The need for computational speed precludes a full co-occurrence matrix and a fair number of feature spaces are mandatory to overcome the overlapping of features between macro-structures. This prompts the use of other descriptors in the second-order statistics. Three features are devised for this research in order to strengthen the clustering power of the algorithm. These are the edgeness, the pixel difference and the roughness.

3.4 *Novel Statistical Texture Features*

Edgeness

The "edgeness" is similar to the altitude contour lines of a landscape map. Each pixel of a subimage is a potential edge-pixel if its gray level is greater than the pixel immediately above it (i.e. a distance of one pixel at a direction of 90°) by a threshold. It is a potential edge if at least two of its 8-neighbours are also similar potential edge pixels. The total of these potential edges within the subimage is the "edgeness". The threshold for the "edgeness" is taken as 10. (The use of edgeness is demonstrated in Chapter 5).

Formally, let $g=(x_{i+1,j} - x_{i,j})$, where g is the difference in gray level between pixels $x_{i+1,j}$ and $x_{i,j}$; $g_{i,j}=1$ if $|x_{i,j} - x_{i+1,j}|>10$ and at least two of its eight neighbours also satisfy the condition $|x_{i,j} - x_{i+1,j}|>10$, and $g=0$ otherwise. The edgeness of a subimage of size 10x10 pixels is

$$\sum_{i=0}^{9}\sum_{j=0}^{9} g_{ij}$$

Roughness

The "roughness" is the total absolute difference of all pixels from the central pixel of the subimage. This descriptor relates closely to standard deviation but it takes into account the position of a pixel relative to the central pixel of the subimage. It is the difference between a subimage pixel and the central pixel, relative to their Euclidean distance. A pixel close to the central pixel is expected to have similar gray

level to the central pixel if the subimage is homogeneous in texture. Formally the "roughness" for the subimages in this thesis is

$$\sum_{i=0}^{9}\sum_{j=0}^{9}\left[\frac{(x_{i,j}-x_{4,4})^2}{(i-4)^2+(j-4)^2}\right]^{\frac{1}{2}}$$

Pixel Difference

The "pixel difference" is the difference in gray levels between two pixels at distance $\delta=(r,\theta)$, where $r=1$ and $\theta=0°$ and 90°. This is similar to a simplified Robert's edge gradient and equivalent to the much more complicated "run difference matrix" devised by Kim et al for classification of abnormal from normal liver [Kim91]. Formally the total pixel difference in each subimage is

$$\sum_{i=0}^{8}\sum_{j=0}^{9}|x_{i+1,j}-x_{i,j}| \; - \; \sum_{i=0}^{9}\sum_{j=0}^{8}|x_{i,j+1}-x_{i,j}|$$

These three statistics are easy to compute and can assess the homogeneity of an image or its subimage. Their cost-effectiveness will be proved by their ability to cluster different macro-structures of the liver ultrasound images.

3.5 Stable Statistical Textures: A New Hypothesis

The ultrasonograms (USGs) of a single organ like liver, heart, uterus have the outstanding feature that the images confirm to fairly uniform macrostructure and shape, unless very extensive abnormality exists. The gross anatomy of the liver can be briefly divided as supporting structures (the surrounding capsule, the hepatic canaliculi) and the liver cells. A typical liver USG shows (a) an anterior capsule (just

below the abdominal wall), (b) background of image showing no structure, (c) hepatocellular cells, (d) posterior capsule just below the diaphragm, and (e) the portal canals (Figs. 1.1, 1.2).

Different structural components have different micro-structures which are represented by the statistical textures, and a single component when turned abnormal would have different textures. Basing on this, my hypothesis is:

> *For images of an organ with fixed anatomy (macro-structure), the statistical texture (micro-structure) would have stable pattern. This texture pattern enables segmentation of the images of that particular organ.*

This hypothesis uses an approach different from that of the conventional pattern recognition but consistent with the one proposed in Section 1.1.3 (Figure 1.6).

Further to this hypothesis, as the statistical texture is of stable pattern, information from a small-sized training set would be adequate for supervised segmentation.

The requisite for the proof of the hypothesis is a training set. Five liver USG images are randomly chosen as the training set. These include three images from normal livers and two images with obvious carcinoma. Subimages of different macro-structures are manually identified and the statistical descriptors assembled. The descriptors are: mean, standard deviation, entropy of the gray levels, gray level difference, run length percentage, edgeness, roughness, and pixel difference. The centroids for the macro-structures of anterior capsule, background, normal cells,

posterior capsule, and tumour cells are then determined. Clustering algorithms using these centroids are then used to segment different liver USGs to prove the hypothesis.

3.6 Centroids of Statistical Texture Descriptors

Examination of the raw data collected from the training set shows that they are not normally distributed. The mean value may not accurately represent the centre of gravity for each category of the set. The centroid is used as the reference point, though the centroid itself would be quite close to the mean. Grouping data into clusters and finding their centroid is just like merging regions by centroid linkage. If a region (or a datum) is less than a threshold from the mean of an existing group, it is included into the existing group and a new mean value computed, otherwise it belongs to a new group [Haralick85]. A region (or datum) is compared to a group rather than its immediate neighbour. The centroid is a better estimation of the centre of gravity for a set of data.

It is already known that there are certain amount of overlaps among the data of different macro-structures of USGs. There is usually a gradual change in gray level from one macro-structure to another and edge between regions is seldom sharp. In other words, the boundaries between different classes of data are fuzzy. A datum may belong to a class at a certain degree and at the same time to another class at a certain degree. This degree of belonging (or membership function) is best assessed by the fuzzy set theory introduced by Zadeh [Zadeh65, Ragade77, Bezdek81, Bellman87]. Hence this research employs fuzzy partition algorithms to find the centroids, although such algorithms are meant for clustering rather than centroid-estimation. The classical fuzzy clustering algorithm is the iterative ISODATA [Ball67] which has been

modified by later workers. This research uses a modified version described by Dunn [Dunn74]. Figure 3.3 shows graphically the centroids of the different descriptors for the macro-structures of liver USGs while Table 4.1 shows their statistics.

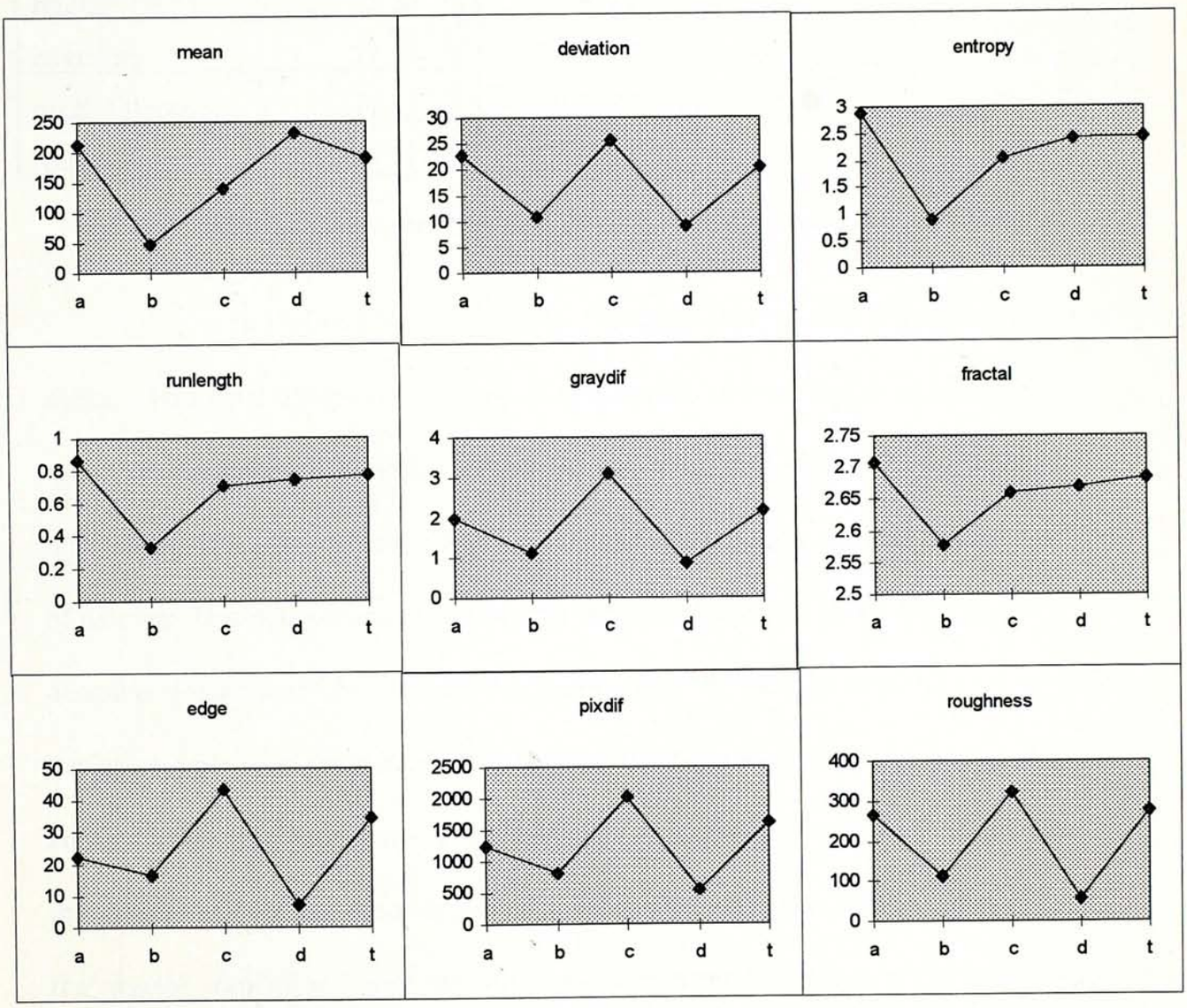


Figure 3.3 Centroids of statistical descriptors from different macro-structures of the training set (a=anterior capsule, b=background, c=cells, d=diaphragmatic capsule, t=tumour)

Descriptor	Mean	Std Deviation	Minimum	Maximum
mean gray level	164.30	74.56	4.10	255.00
deviation	18.19	13.79	0.00	84.60
entropy	2.13	0.87	0.00	3.72
run lrngth	0.68	0.25	0.00	0.99
gray difference	1.90	1.46	0.00	6.88
fractal	2.66	0.17	1.83	3.04
edgeness	25.96	30.72	0.00	125.00
pixel difference	1277.81	1033.34	0.00	5198.00
roughness	216.71	334.92	0.00	2408.68

Table 3.1 Statistics of the descriptiors from the different macro-structures of the training set

In Figure 3.3 the curves for the descriptors clearly show three types of shape. The fractal dimension, entropy of gray levels, and run length percentage are of one type; the standard deviation of gray levels, gray level difference, edgeness, pixel difference and roughness are of another type; and the mean gray level is the third type of its own. This phenomenon suggests that statistical features of the same curve shape describe similar characteristics of the image. The "w"-shaped descriptors (standard deviation, gray level difference, etc.) describe the difference between a pixel value and another value (the latter may be another pixel at a certain position, or, in the case of the standard deviation, the average value of the region). The entropy of gray levels, the fractal dimension, and the run length percentage measure the distribution equilibrium of the surface. Of course, there is no similar descriptor as the mean gray level in the present feature spaces.

In the next chapter, we shall examine whether these curve shapes exist in other liver USGs, whether the shapes are stable (as hypothesized), and whether we can use these particular shapes to segment images.

CHAPTER 4

NORMAL LIVER IMAGES

The hypothesis stated in Chapter 3 (Section 3.5) has two parts: (a) a stable pattern of statistical texture, and (b) an effective segmentation algorithm derived. Accordingly, the proof has two approaches, each at different levels of complexity. The first is to assemble statistical texture parameters from a set of images from normal liver. If these parameters have the same or closely similar shapes as the training set, the first part of the hypothesis is not to be rejected. The second approach is to derive an algorithm, *from the shape of the curves*, to cluster the images. If the hypothesis is true and viable, images should be successfully clustered.

4.1 Further Description of Normal Liver USG

With reference to the example of an ultrasonic image of the normal liver in Figure 1.1, the gray scale ultrasonic appearance of normal liver cells is a homogeneous internal architecture with fine uniform distribution of echoes without focal distribution [Green77, Weill87]. The supporting tissues (anterior capsule, bile ducts, posterior capsule) are of higher gray levels (brighter). The transition from cell to supporting tissue (or vice versa) is a zone of gradual (or fuzzy) change of gray levels although the core of the latter may have relatively sharp change. The gradual transition is more marked with the posterior capsule, so much so that a human (especially one without medical knowledge) may find it difficult to classify some subimages. This is due to the diffraction of sound waves into different directions from the broadly curved posterior capsule.

Looking at the image (we must note that the image is not equalized) , one may expect that:

(a) the background is the darkest region and the gray scale is more uniform

(b) the capsules are the brightest areas with largest variation in the brightness

(c) the brightness for cells occupies an intermediate position.

Taking the mean gray level to represent brightness and the standard deviation to represent homogeneity, we may expect a "✓"-shaped curve. Indeed this is so. When the subimages of different macro-structures are manually identified and the statistical descriptors grouped (in the same manner as for the training set), the same curve-shape is obtained (Figure 4.1). From the curves, one can imagine that a threshold will successfully divide the image into two groups: capsules (anterior and posterior) and noncapsule (background and cells). The new statistical parameter of this thesis — the edgeness applies such a threshold (c.f. Section 3.4 and Section 5.2.2).

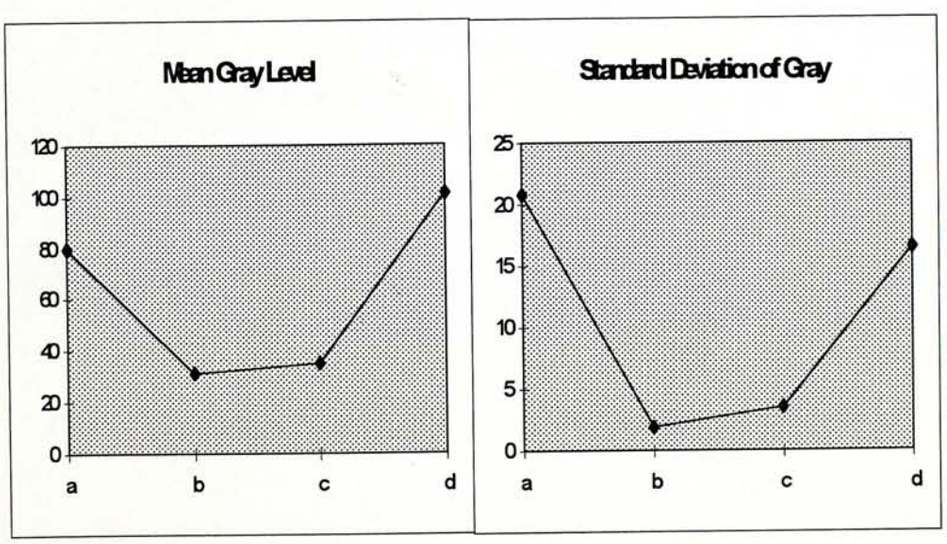


Figure 4.1 Curves of two statistical parameters in an unequalized normal liver USG (a=anterior capsule, b=background, c=cell, d=posterior capsule)

4.1.1. Equalized images

Figure 4.2 shows the same image of Figure 1.1 after gray level histogram equalization. It is obvious that the liver cells are no longer as homogeneous as before and show much more variation in "brightness". Instead of being "squeezed" around the maxima of the gray scale, they are "stretched" out to occupy a broader strip of gray scale. The cells subsequently have higher standard deviation than the capsules in the training set. The borders of the capsule become even more fuzzy.

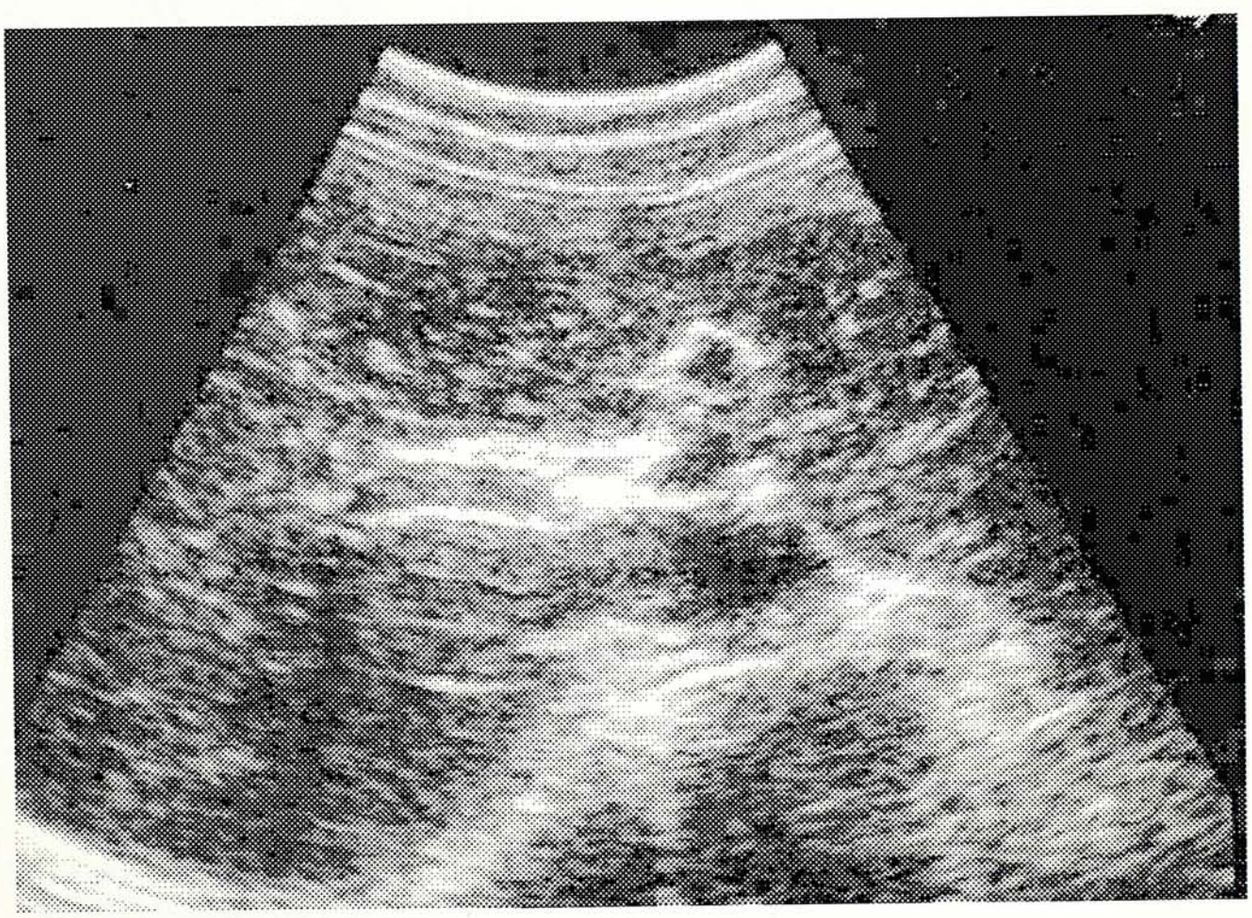


Figure 4.2 Image of Figure 1.1 after gray level histogram equalization

4.2 Stable Statistical Descriptors in Normal Liver Images

Ten ultrasound images of normal liver were randomly taken and statistical texture features were extracted in the same manner as for the training set. Subimages

of different macro-structures were manually identified and the statistical descriptors grouped. These were then graphically plotted (Figure 4.3). The curves of the individual descriptors are very similar to those of the training set (Figure 3.3), supporting the hypothesis that statistical textures conform to a stable pattern.

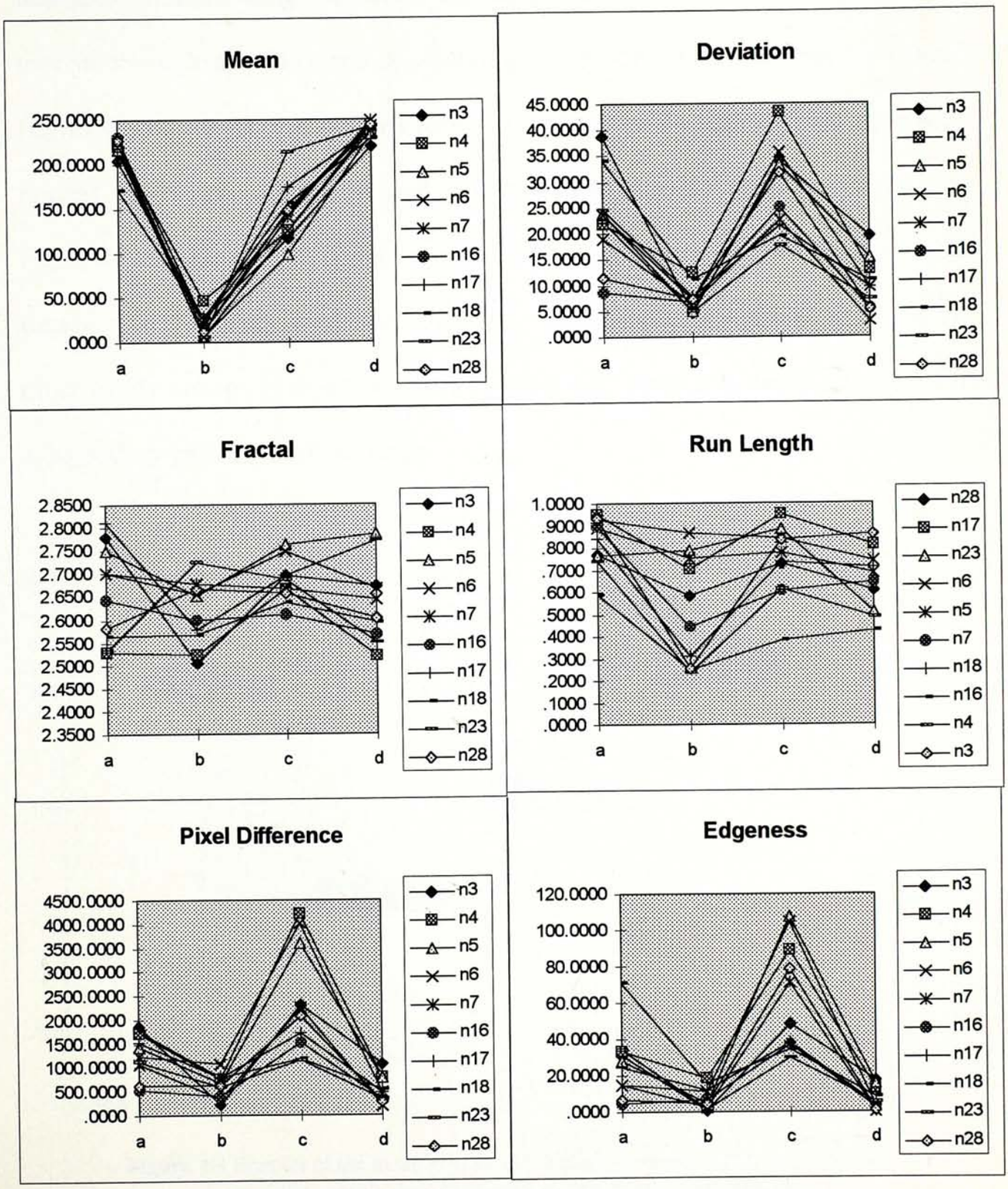


Figure 4.3 Some statistical features extracted from subimages of USGs
(a=anterior capsule, b=background, c=cell, d=posterior capsule)

Close examination of Figure 4.3 shows that some descriptors are more stable. The mean gray level is most stable, even in absolute values. There is so little variation among the images that the curves of the means can nearly overlap into a single curve. This explains the findings of previous researches that the gray level is the best descriptor in segmenting monochrome scans [Coleman79], discriminating USGs of liver tumours [Youssef89], and in separating images of fatty livers from the normal [Layer90]. But we must recall that these researches worked with selected regions of interest which were usually the centres of the regions, avoiding the fuzzy borders. Figure 4.4 is a boxplot of the mean gray levels taken from one of the testing images (image #n3). It clearly shows that there are overlaps among capsules and cells. In other words, though itself an excellent discriminator, mean gray levels alone cannot adequately segment the whole image.

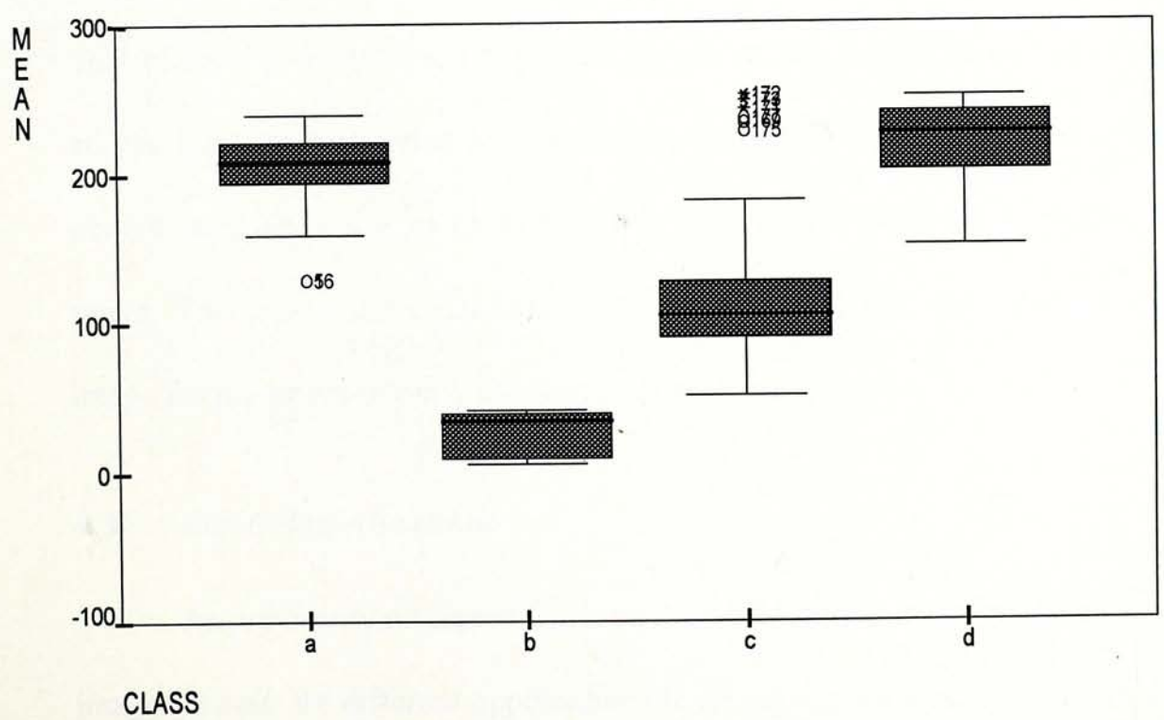


Figure 4.4 Boxplot of the mean gray levels of macro-structures of a normal USG (a=anterior capsule, b=background, c=cell, d=posterior capsule)

Other statistical parameters have the same curve-shapes as the training set except the fractal dimension and the run length percentage. The curves of the latter two parameters are inconsistent. These two parameters have the same property of measuring the surface contours of an area rather than the spatial relationship between two points of an area as the other second-order statistics. The clustering algorithm will not include these two parameters. (For the sake of experimentation, the research did try these two parameters in the clustering algorithm. As expected, the result was unsatisfactory.)

For the other parameters, the curves from different images are very close. (The parameters for the cells of image #n4 are consistently higher than those of other images, most probably due to variation in manual isolation of subimages for the cells.) The absolute values of each parameter differ among different images, depending most likely on the image conditions. However, they conform to a very similar shape, not only of their own but also of the training set. It is not difficult to visualize that the curves may nearly overlap by shifting the individuals upwards or downwards. The curves themselves are so close that this vertical shift varies only within a narrow range. This observation supports the hypothesis that the statistical textures of the image from a single-organ USGs have constant pattern.

4.3 Clustering Algorithm

As previously discussed (Section 2.1) clustering is more suitable for ultrasonic images. There are different approaches for clustering, each one having its theoretical and practical merits. We already have built the centroids for the training set and it is most convenient, also fastest, to cluster a subimage according to the distances of its

statistical descriptors from these centroids. The *K*-means clustering[8] is fast, simple and well established, and is therefore used for this research. The same clustering method was used by many previous researches [Lerski79, Mailloux84, Momenan88, Botros88].

The algorithm has two steps. In the initial step, the mean gray level, with its most unique and discriminating curve shape, clusters the macro-structures into background, cells and capsules. The anterior and posterior capsule are grouped as one category as they share very close statistical features, and they can easily be separated by their position in the image. This also speeds up the algorithm.

There are certainly some mis-classified subimages after this initial clustering. Figure 4.4 illustrates not only such presence but also that the most mis-classification (or overlap) occurs between capsules and cells. This is also obvious if we refer back to our ultrasonic images.

The second step of the algorithm is essentially the correction of this mis-classification. Each subimage is tested with other descriptors to see if it should remain as classified as such or be re-allocated to another cluster. We may expect that relatively heavier computation is required to differentiate liver cells from the capsules than from the background. Indeed it is so. Pixel difference alone is adequate to re-allocate mis-classified background into cells; gray level difference and entropy are required to differentiate cells into background or capsule; and three descriptors (pixel difference, roughness, and gray level difference) are required to discriminate between capsules and cells. Figure 4.5 shows this algorithm.

---

[8] *K*-means is based on the minimisation of a performance index which is defined as the sum of the squared (Euclidean) distances from the points under consideration to a reference point [Tou74].

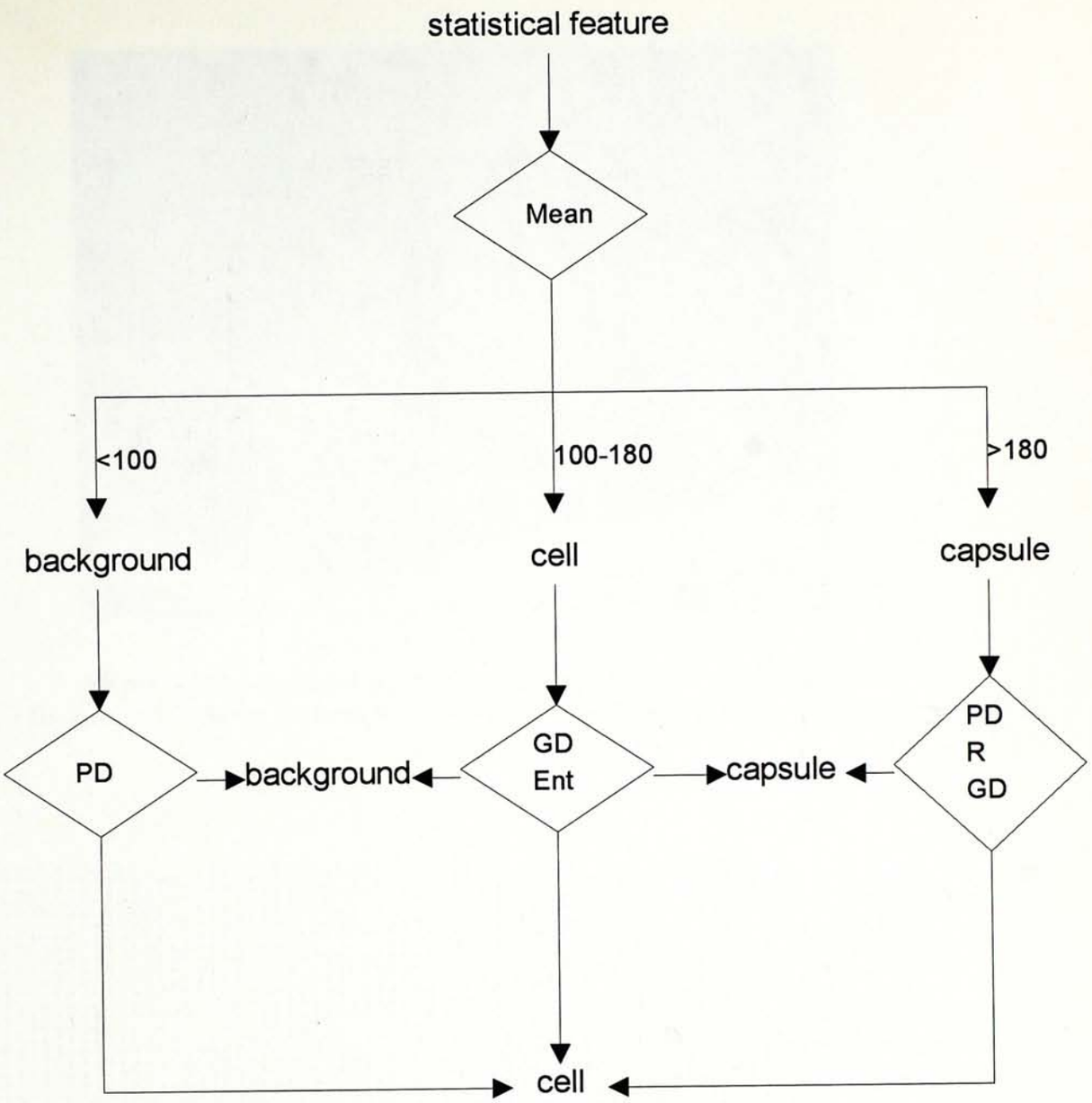


Figure 4.5 Segmentation algorithm used in the study
(Ent = entropy, GD = grey difference, PD = pixel difference, R = roughness)

A result of such clustering is illustrated in Figure 4.6 (an equalized liver USG), Figure 4.7a (the digitized result after clustering), and Figure 4.7b (which converts Figure 4.7a into the corresponding gray scales).

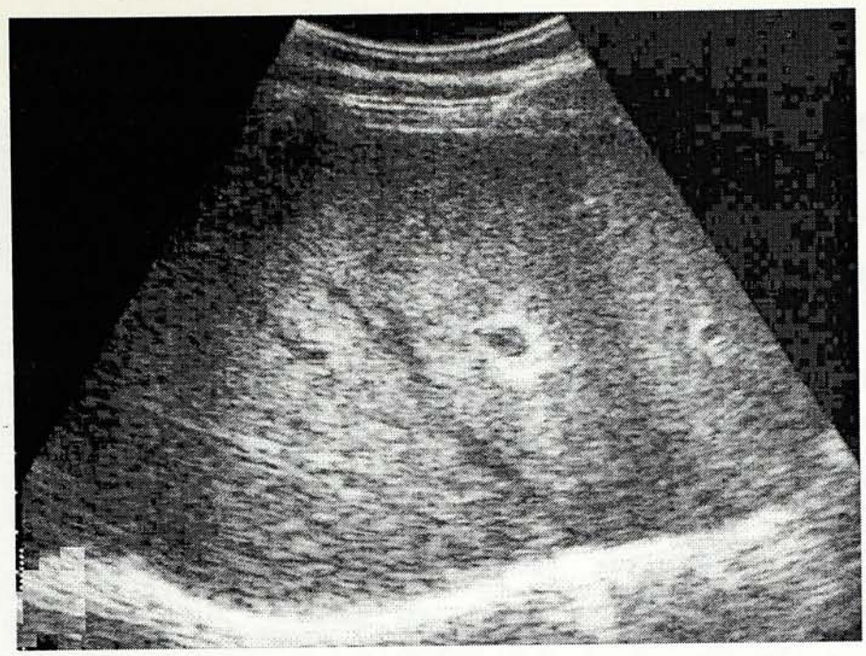


Figure 4.6 The equalised (grey level histogram) image of a liver USG
(The bottom left corner has been artificially touched to enforce a posterior capsule)

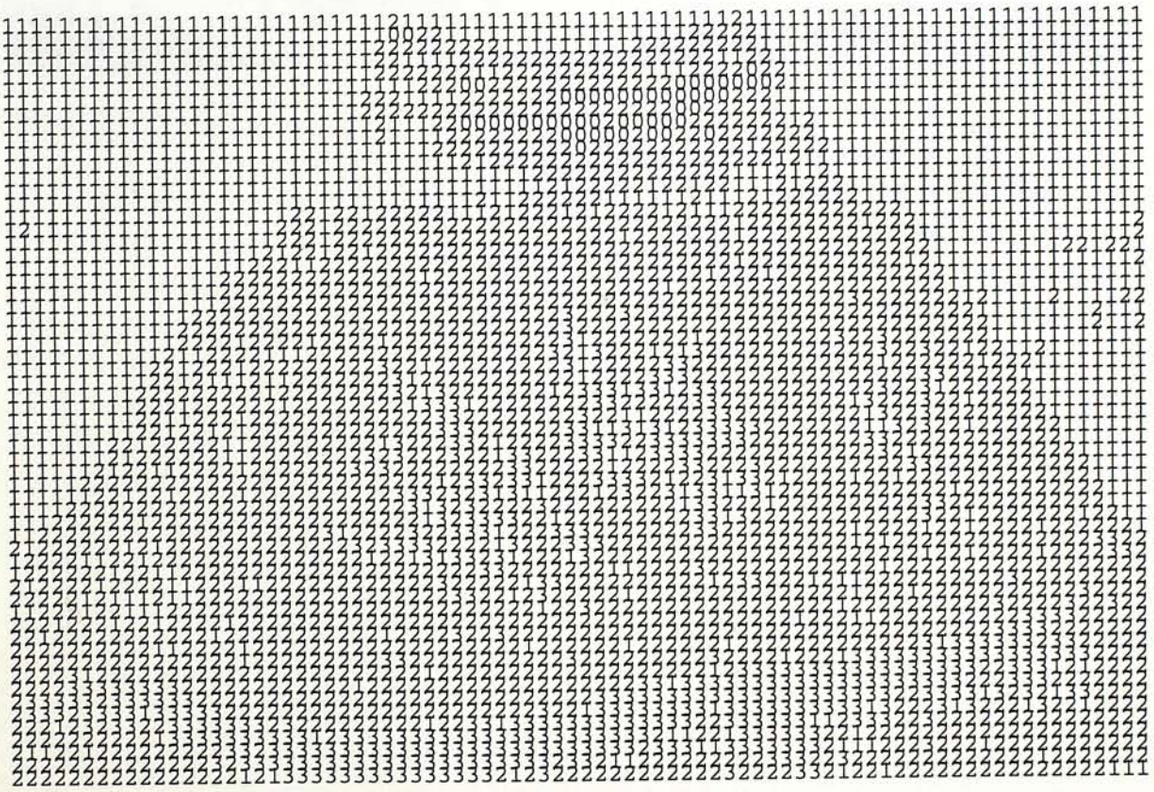


Figure 4.7A Digitized result of Figure 4.6 after clustering
(0=anterior capsule, 1=background, 2=cell, 3=posterior capsule)

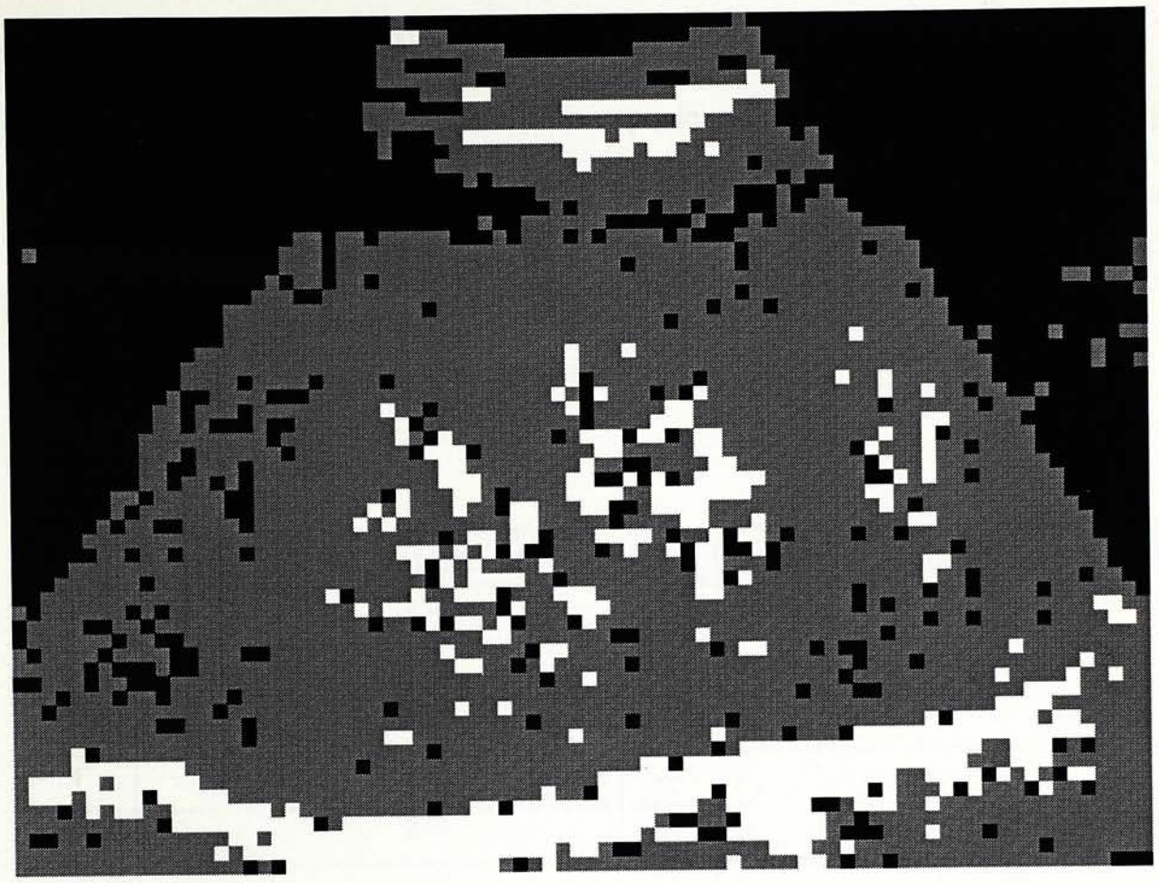


Figure 4.7B Grayscaled result of Figure 4.6 after clustering [9]
(capsule: gray level 255, background: gray level 0; normal cells: gray level 100)

[9] A digitized map such as Figure 4.7a is visually less informative than a grayscaled image such as Figure 4.7b. However, numerical data maps are the raw outputs of clustering and they enable localization of corresponding subimages to check the results. Only numerical data maps will be presented for the rest of the thesis.

4.3.1. Accuracy of the algorithm

The algorithm has been tested on 20 liver USGs (the first 10 are those in Section 4.2). The digitized segments visually represent the images closely. To test the accuracy of the results, the anterior capsule subimages of the testing images are manually identified (without knowledge of the results of the algorithm). The numbers of anterior capsule clustered by the algorithm are then compared with those manually identified. Table 4.1 shows the comparison.

USG	number of subimages isolated by algorithm	number of subimages identified manually
1	258	313
2	234	248
3	283	248
4	162	155
5	249	177
6	177	126
7	16	44
8	313	280
9	154	188
10	358	332
11	149	218
12	48	176
13	249	196
14	328	318
15	228	211
16	99	112
17	210	272
18	284	150
19	284	273
20	204	257

Table 4.1 Number of subimages identified as anterior capsule

The average numbers of subimages in Table 4.1 as identified by the algorithm and manually are 214.4 and 214.7 respectively. Disregarding the USG #7 which has a peculiarly low value, the average numbers are 224.8 and 223.7 respectively. To see the accuracy of localizing the sites of the anterior capsules, five images from Table 4.1 were randomly chosen. The sites of the anterior capsule identified by the algorithm and manually were compared. Table 4.2 shows the result. Figure 4.9 is the digitized result of the best match (#11). Both methods correlate very closely, showing that the algorithm's performance is as accurate as a particular human expert.

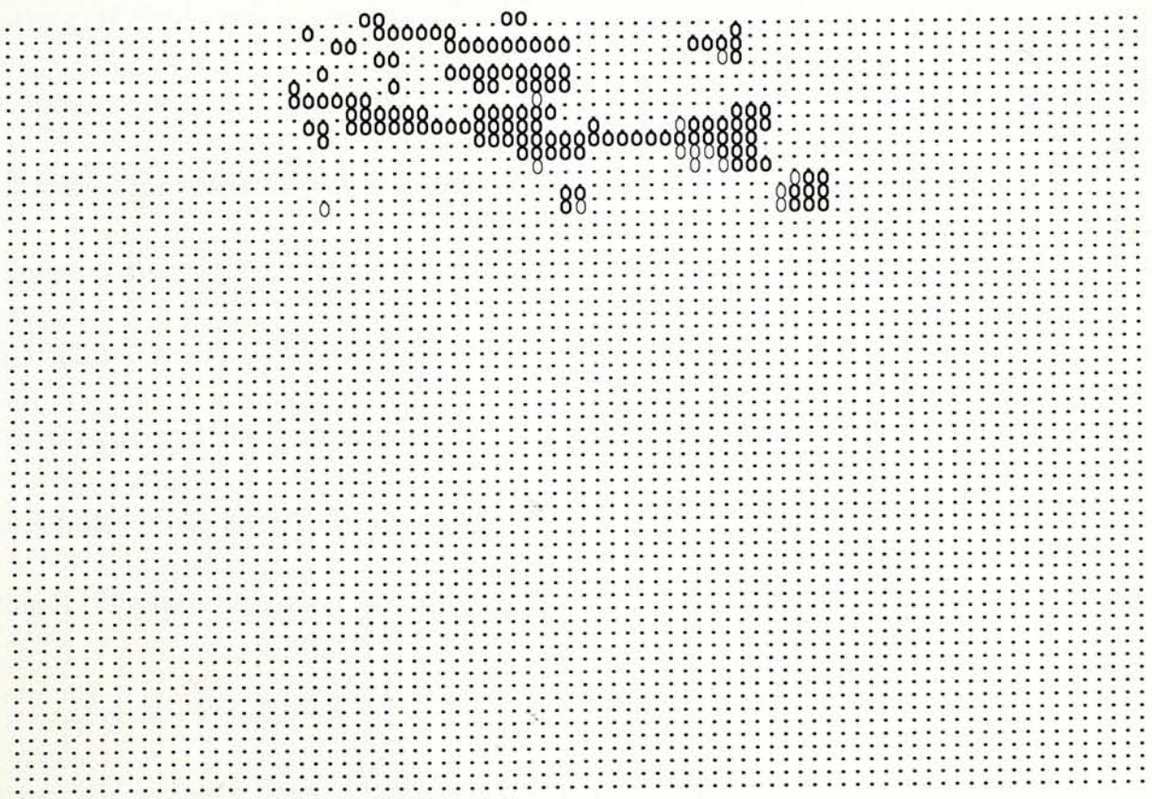


Figure 4.8 Subimages of anterior capsules (#11, Table 4.2) agreed by both methods of identification.
(0=anterior capsule, subimages agreed are shown in bold type)

USG (Table 4.1)	Number of subimages identified by algorithm	Number of subimages agreed on manual identification	Percentage of agreement
1	258	194	75.19
3	283	217	70.67
8	311	297	81.67
11	149	134	89.90
17	210	178	84.76

Table 4.2 Agreement of localizing the sites of the anterior capsules

4.3.2 The algorithm and ultrasound artifacts

The ultrasound artifacts described in Section 1.1.2 have no characteristic features of their own. They are identified with reference to a macro-structure or an abnormal structure in the USG. How would the algorithm of this thesis handle these artifacts? This research has not included ultrasound artifacts in the training set. Even if these artifacts have been included, they would just share some features of those macrostructures most close by. The shadows look like the background (at least to the human eye) and the enhancements look like the capsule. They would be classified accordingly. The algorithm gives results which truly reflect the visual effects of the images. The interpretation of these results needs more maneuver.

Figure 4.9 shows an USG image with a small cyst close to the anterior capsule. There is a "tail" of enhancement (of higher gray level than the surrounding cells) extending right to the posterior capsule. Figure 4.10 is the resultant digitized map after clustering. The enhancement was classified as posterior capsule by the algorithm. Considering the texture, one would accept this classification. But, of course, considering the shape and position of the region, one would classify it as a new non-cell, non-capsule region.

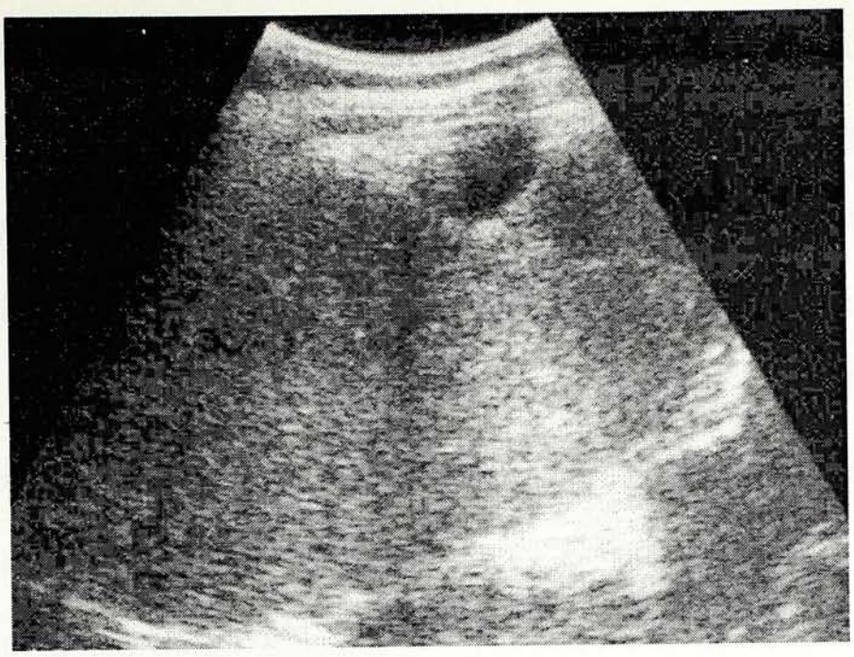


Figure 4.9 Equalised image of liver with a cyst near the anterior capsule (just right to the midline)

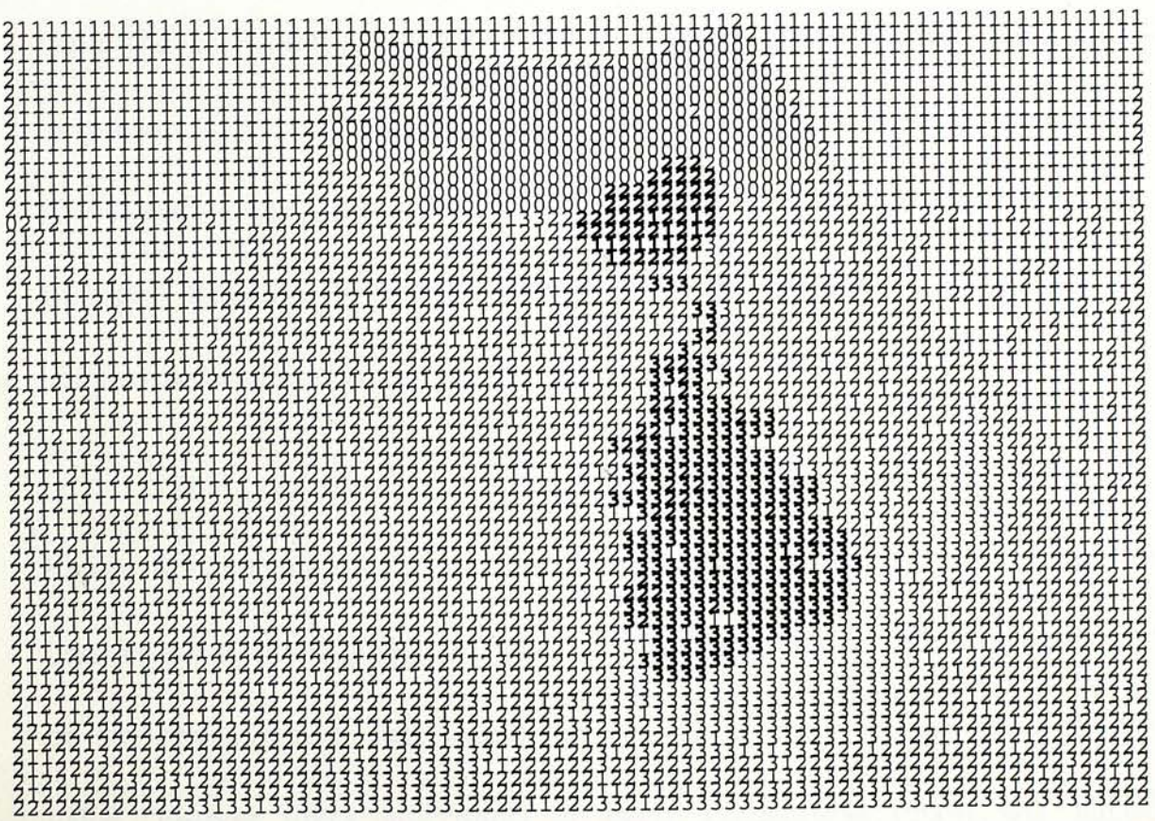


Figure 4.10 Digitized map of Figure 4.9 after clustering (the cyst and enhancement are shown in bold type)

4.3.3 Fuzzy algorithm for clustering

Since the USGs have fuzzy borders between their macro-structures and lost of overlaps among the statistical parameters, it is a logical strategy to apply fuzzy algorithms to cluster the image rather than the decision-tree method as depicted in Figure 4.5. Indeed this thesis had tried fuzzy clustering using algorithms described by Bezdek [Bezdek81]. The parameters used were the same as those in Figure 4.5. The algorithm worked even faster as an subimage was compared against a set of centroids rather than different centroids in steps. However, the clustering was less satisfactory with lots of mis-classifications. The explanation may lie on the extensive overlap among the parameters. When there are two or three parameters, fuzzy algorithm may correctly classify a subimage. But when there are six parameters, the extensive overlaps may interfere each other and confuse the fuzzy membership. On the other hand, the decision-tree method makes use of the discriminating power of the mean gray level of the subimage for initial clustering, and then different other parameters, few at each phase, to re-adjust the overlaps. In this sense, the decision-tree method simulates iteration for stability (using different parameters at different phases) to eliminates the effects of the overlapping areas.

There remains one question: if fuzzy algorithm is used in the decision-tree of Figure 4.5 instead of the shortest Euclidean distance classifier, would the result become even better? This thesis has not pursuited this side-track because should there be any improvement, that would most likely be more of theoretical than practical value. Putting fuzzy algorithms in every comparison phase will lengthen the computation time.

4.4 *Evaluation of the Algorithm*

The algorithm is certainly simple and fast, and it does its job nicely. But there are obvious deficiencies from Figure 4.7 and Figure 4.10. Some subimages in the region belonging to the background are classified as cells. Heuristically, one will regard these subimages as mis-classifications. We must recall that the algorithm works on equalized gray levels. While equalization evens out brightness and brings out finer details, it may also introduce noise. An USG has the mean gray level for the whole image around 40-50 before equalization and around 125 after equalization. The mean gray level for the background and the cells are very close (30-40) before equalization; hence some mis-classification may result after equalization of gray levels. The mis-classification is mainly the side-effects of operation with second-order statistics because equalization generates more variation among pixel values within a subimage.

For better segmentation, the algorithm needs methods to eliminate the effects of noises. There are two approaches for this. The first one is to compare the regions clustered from the unequalized images. This approaches has obviously many defects. It doubles the computation, and needs training sets for raw images which depend unpredictably on the ultrasound operator and scanner. The second approach is to bring in neighbourhood relationship. A subimage is to be grouped into the same cluster as the majority of its neighbour. For example, if a subimage labeled as "cell" is totally surrounded by subimages labeled as "background", it is then clustered as "background". In such case, we need to define what is the majority. In the next Chapter, we shall see why and how these approaches are used to refine the detection of subimages suspicious of carcinoma of liver.

CHAPTER 5

IMAGES OF LIVER CARCINOMA

5.1 Characteristics of Liver Carcinoma

All malignant tumours share a common characteristic: irregularity (irregular borders, irregular shape, irregular appearance). A carcinoma spreads outwards like the legs of a crab ("cancer" means crab in Greek). The boundary between normal and tumour cells is usually indefinite. The fuzzy boundary together with nonspecific structures (e.g. necrosis, calcification) complicating the texture organization of tumour make precise segmentation of abnormal lesions in ultrasonic images difficult [Mailloux84]. A tumour can be either less or more echogeneic than normal tissue making the mean gray level an unsatisfactory descriptor for tumour.

Carcinoma of liver arises from liver cells, hence "hepatocellular" carcinoma ("hepato" means liver). The problem of automated detection of liver carcinoma is the differentiation of abnormal cells from normal ones. Referring back to the statistical descriptors of the training set (Figure 3.3), there are two important observations:

(a) For entropy, fractal and run length percentage, the tumour cells occupy the top of the curve, i.e., they are in the most unstable equilibrium. This reflects the irregularity of a carcinoma and is supported by the cellular pathology of tumors that they consist of rapidly dividing cells.

(b) For other descriptors, those for homogeneity of textures like pixel difference, gray level difference, standard deviation, the tumour cells occupy a position between

the normal cells and the capsule. This is again expected as tumour arise from normal cells. So their descriptors are closer to normal cells than to capsules.

These lead to the conclusion that: *if the position of the capsule in the curves of these descriptors can be located, the position of the tumour cells along the same curve can be estimated.*

5.2 *Algorithm for Tumour Detection*

The algorithm continues from the one described in Chapter 4. Tumour cells, if present, would be mis-classified into normal cells or into the capsule (if the tumour cells are hyperechoic). As we see from Figure 4.3, though the descriptor curves of the images are very close, they may shift slightly in vertical direction. Fixed reference centres may have larger variation for tumour centre which flows between the centres for normal cells and capsules. As the tumour centre depends on the relative positions of the macro-structures in the distribution curve of a particular descriptor in a particular image, the centroid values of the training set is not used. This section of the algorithm starts with the determination of descriptor for the capsule in *the liver image under consideration*. This approach has the advantage that the algorithm is dynamic, adjusting itself for the shifting of descriptor curves of the image under consideration up or down away from the training images. The crucial point is then the accurate identification of the capsules.

There are three problems to be solved before implementing the algorithm:

(a) Which statistical descriptors to use?

(b) How to isolate the capsule subimages?

(c) How to estimate the position of the tumour cells in the descriptor curve?

5.2.1 Which statistical descriptors to use?

There are three descriptors having a "✓" (the check sign) curve shape: fractal, entropy, and run length. Fractal dimension has been discussed in Chapter 3 and is found to be variant in different images. Chapter 4 (Figure 4.3) demonstrates the inconsistency of the run length percentage as well as the fractal dimension. So the entropy of gray levels (ENTGL) remains the best (and the only) choice. Figure 5.1 shows the curves for ENTGL from the images studied in Chapter 4. Except for two curves, the shape of the curves are quite consistent.

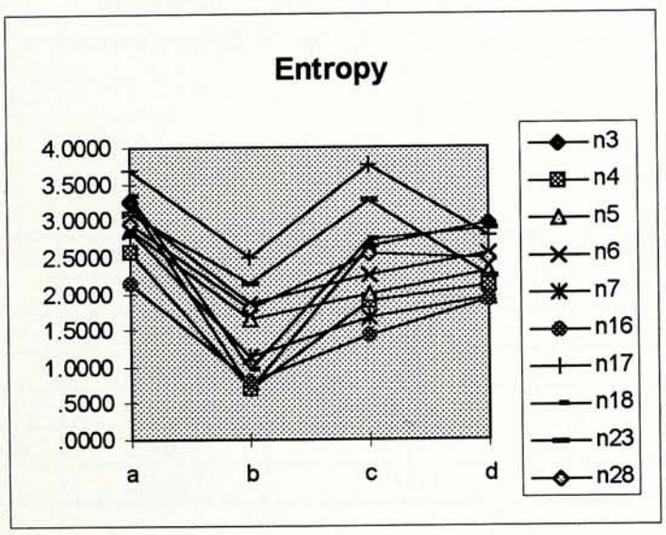


Figure 5.1 Entropy of gray levels in 10 normal USGs
(a=anterior capsule, b=background, c=cell, d=posterior capsule)

In order to substantiate this observation, eight more normal USGs in succession are sampled and the entropy values of manually selected subimages plotted. Again the shape of the curve is quite constant (Figure 5.2). This observation

is further supported by statistics from the training set in Chapter 3 (Table 5.1). So entropy was chosen as a descriptor to be used.

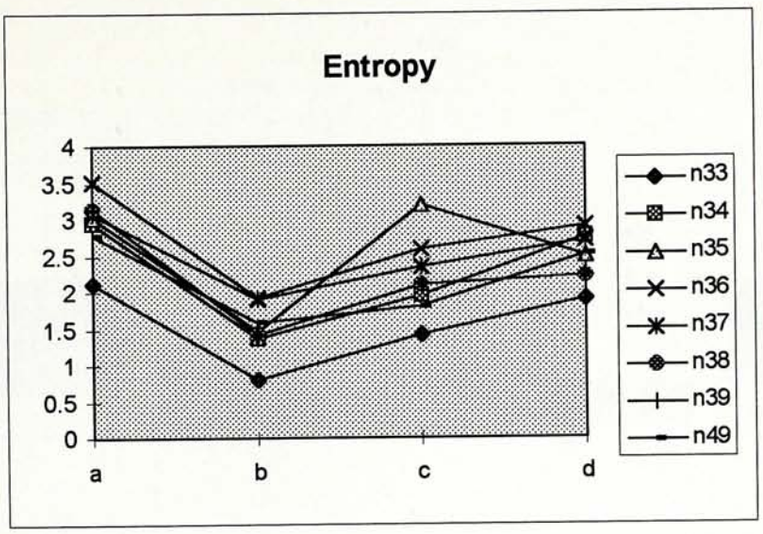


Figure 5.2 Entropy of gray levels of eight normal USGs
(a=anterior capsule, b=background, c=cell, d=posterior capsule)

	Entropy of Gray Levels		Run Length Percentage	
	mean	standard deviation	mean	standard deviation
anterior capsule	2.8939	0.3959	0.8599	0.0948
background	0.9032	0.6567	0.3318	0.2116
cells	2.0519	0.5071	0.7038	0.1878
posterior capsule	2.4026	0.6545	0.7413	0.2003
tumour	2.4317	0.5114	0.7685	0.1769

Table 5.1 Entropy of gray levels and run length percentage of the training set in Chapter 3

Another descriptor needs to be chosen from the "w"-shaped descriptors: gray level difference, pixel difference, standard deviation, etc. The choice was finally made on the gray level difference when the statistics of the descriptors in the training set were further examined. The gray level difference shows the least overlaps between normal cells and tumour, and between tumour cells and capsules. (For experimen-

tation, the pixel difference was also tried in the algorithm and, as expected, the result was less satisfactory than that given by gray level difference.)

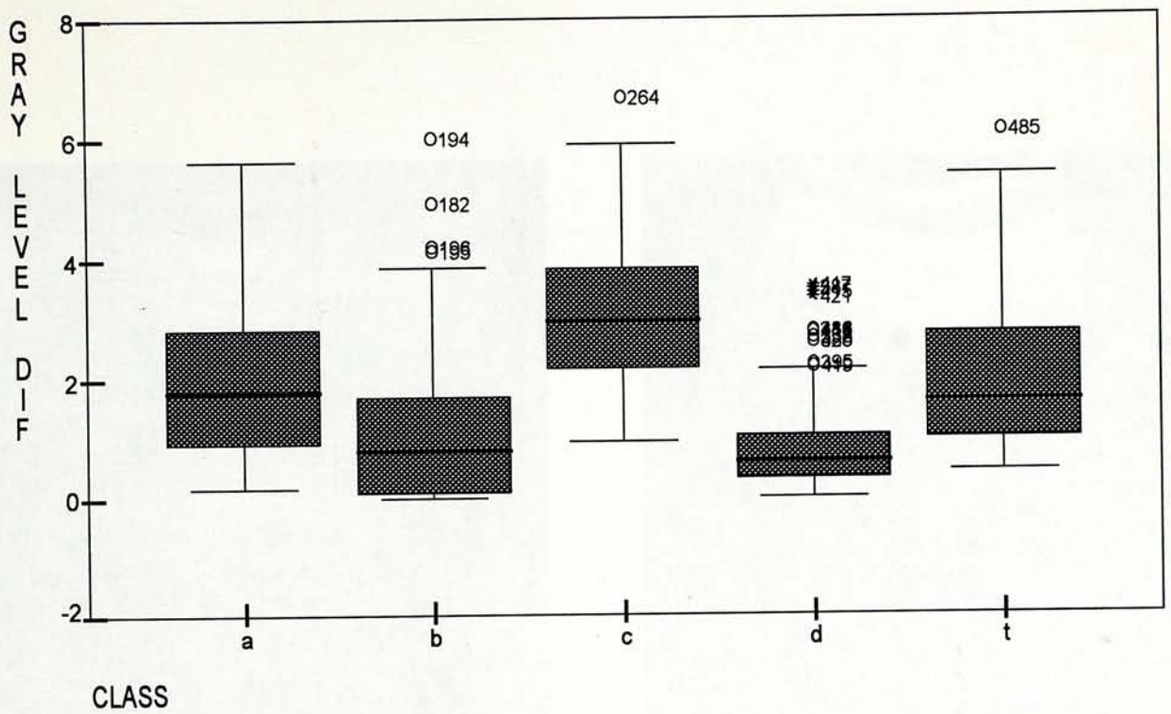


Figure 5.3 Boxplot of gray level difference of the training set in Chapter 3 (a=anterior capsule, b=background, c=cell, d=posterior capsule, t=tumour)

5.2.2 How to isolate the capsules subimages?

At first sight, this question seems redundant. The capsules have been successfully isolated from the first part of the algorithm as detailed in Chapter 4. We must recall that the algorithm works on data drawn from images with equalized gray level histogram. The beauty of the equalization is to show up otherwise hidden objects in the images by spreading out the gray levels uniformly. It is extremely helpful to bring up tumour cells amidst the normal cells (Figs. 5.4, 5.5.). However, it also intensifies the subimages bordering the capsule regions and mis-classify them into capsules. This will be a source of inaccuracy in determining the centroid for the capsule

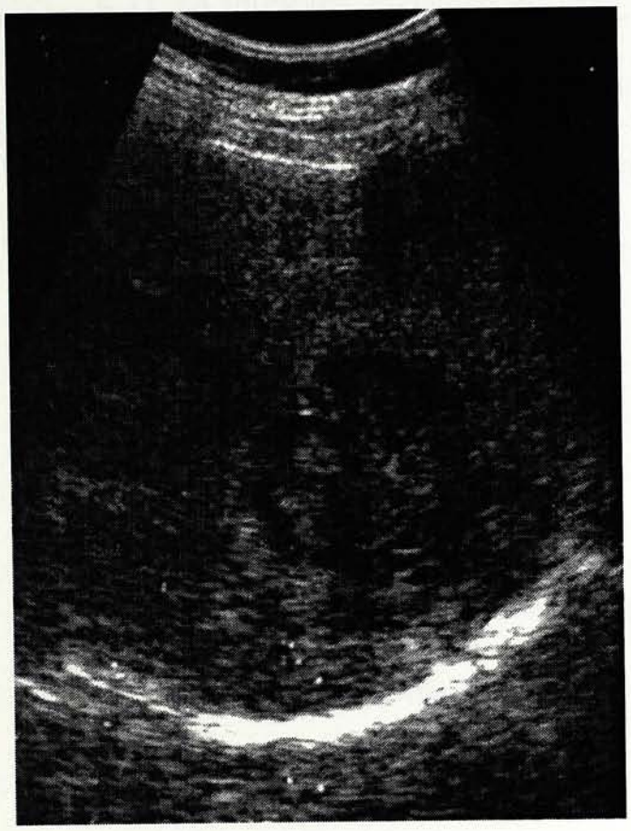


Figure 5.4 A liver USG showing a carcinoma

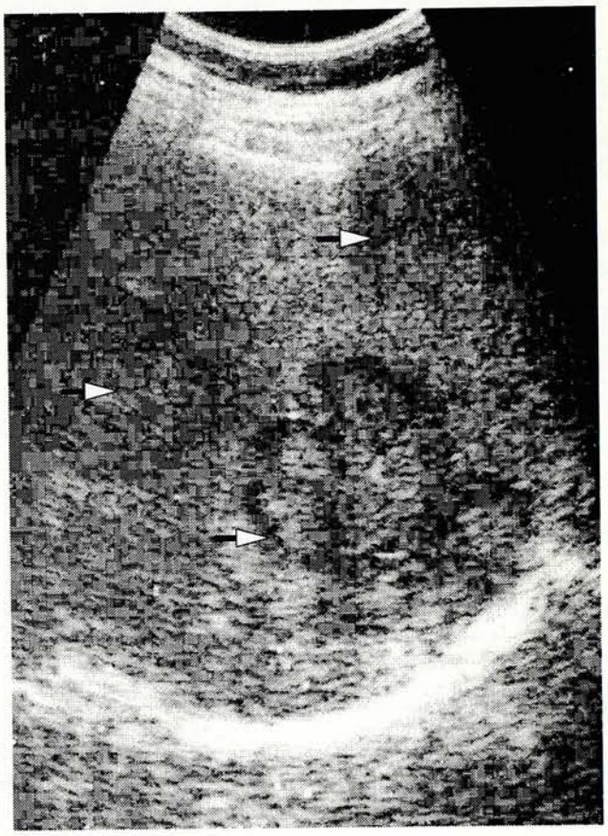


Figure 5.5 Image of Figure 5.4 after equalisation of gray level histogram (showing three tumour deposits indicated by white arrow-heads)

The capsule has the highest acoustic impedance, hence the highest gray level. The contrast between capsule and non-capsule structures is highest before gray level equalization. So the original image is used for isolation of the capsule. A tempting approach is to use a threshold of the gay level for isolation of the capsule (Figure 4.1). (A threshold of the mean gray level of the image plus one standard deviation was tried in the research.) But the amount of capsule in an image varies, influencing the overall gray value of the whole image. Another source of inaccuracy is that the ultrasono-grapher may adjust the brightness and contrast level during scanning, increasing the gray levels of the cells and hence their weight in the overall mean level of the image. A more invariant approach is required.

The edgeness described in Chapter 3 is an invariant descriptor as it computes a difference between two pixels at a fixed direction and distance; it disregards absolute values. If the threshold for the difference is appropriately set, the fuzzy borders of the capsules (which could well be cells with higher gray levels due to reflection of sound waves from the neighbouring capsule) can be eliminated, giving a better estimate of the capsule subimages. So the edgeness was used for isolating the capsule subimages from the original image.

Figs. 5.6 and 5.7 show the results of the two methods of isolating the capsule subimages in digitized forms. Figure 5.8 shows the result using the conventional Sobel edge detector (those subimages with Sobel gradients greater 10 are represented by "1", else by "0"). It is obvious on reference to the original image (Figure 5.12) that the edgeness is the most effective and accurate descriptor.

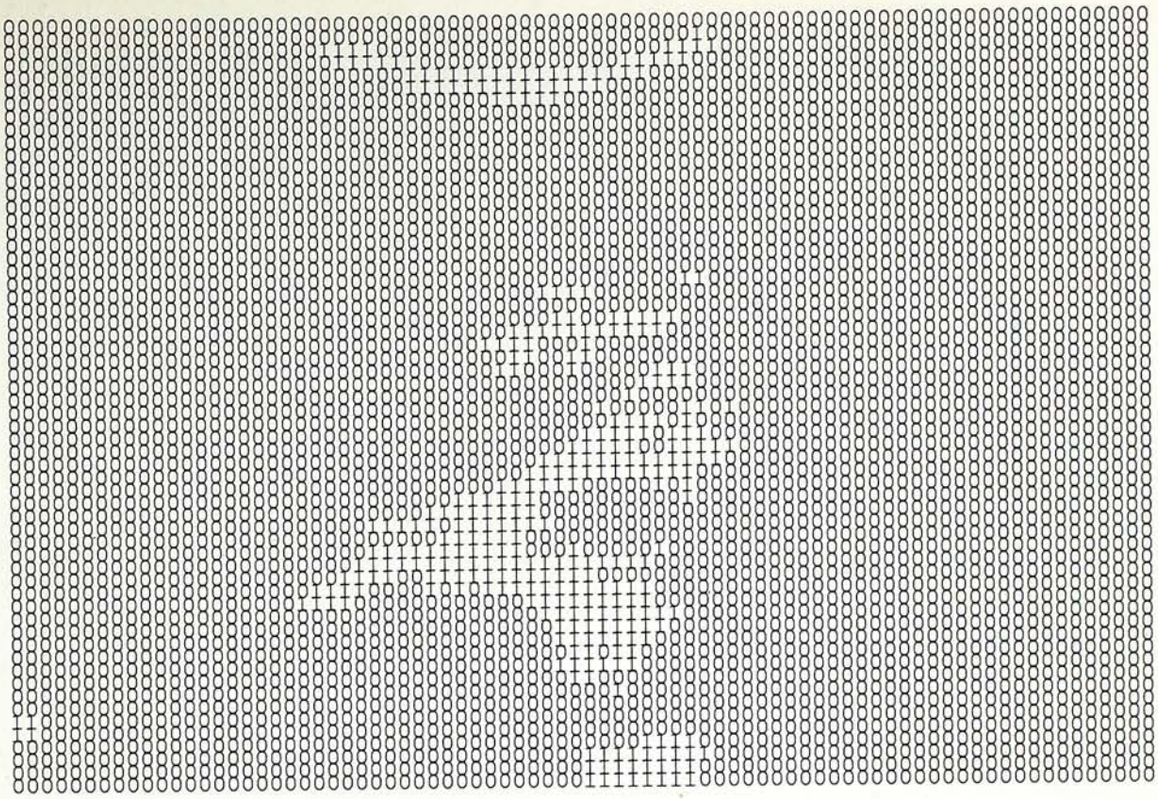


Figure 5.6 Capsule subimages isolated by taking the mean gray level of the whole image
number of subimages isolated = 487
(the USG image is shown on Figure 5.12)

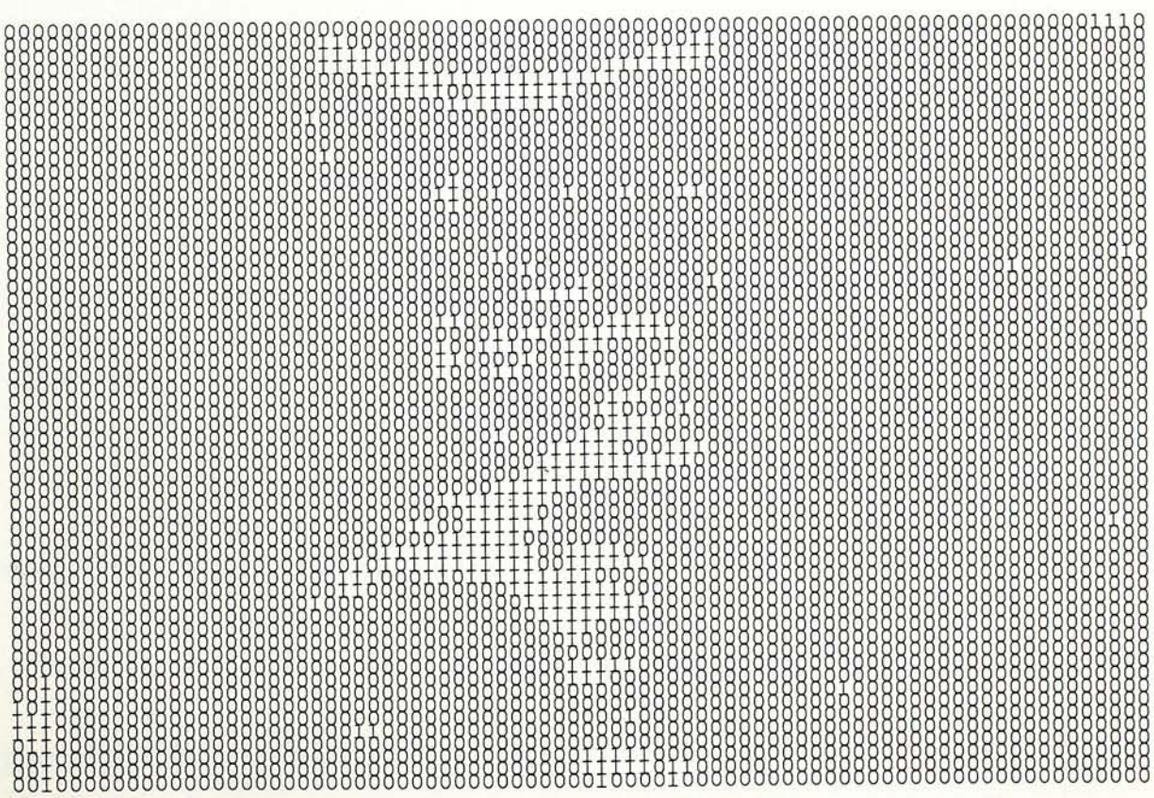


Figure 5.7 Capsule subimages isolated by taking the edgeness of the subimage
number of subimages = 245
(the USG image is shown on Figure 5.12)

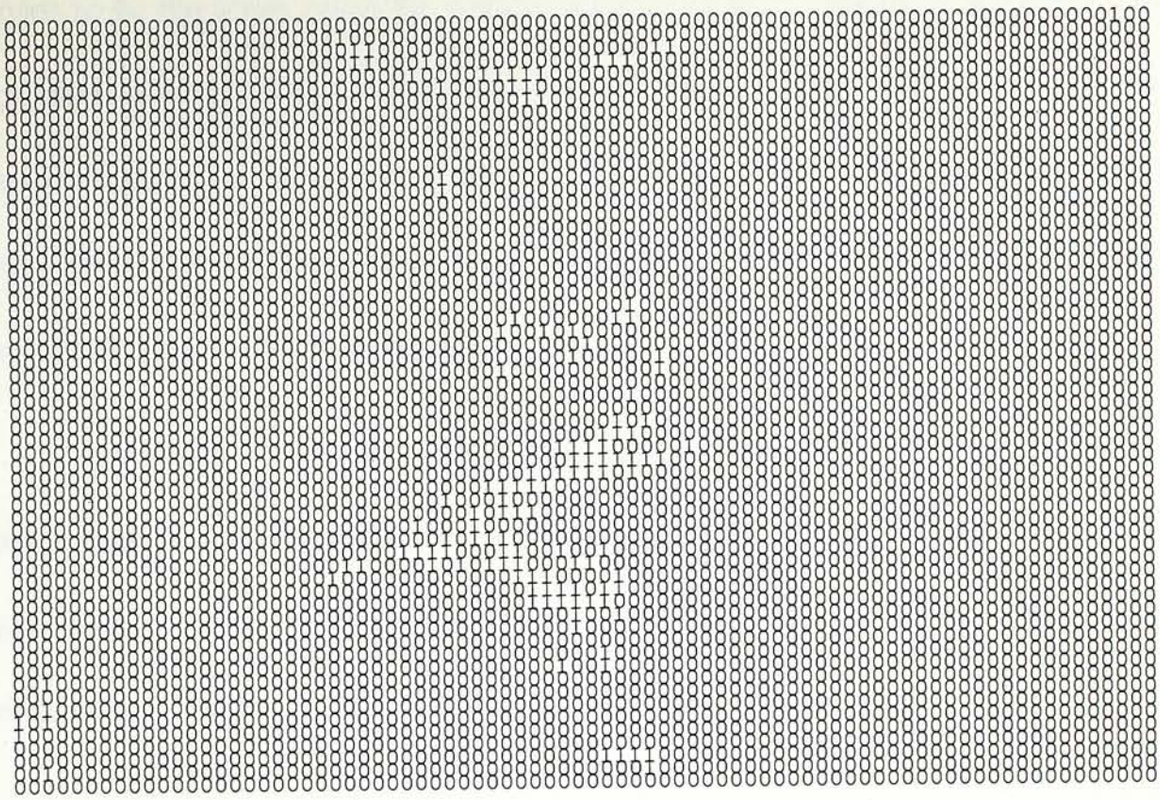


Figure 5.8 Capsule subimages isolated by taking the sobel gradient (same image as Figure 5.6)
number of subimages = 129
(the USG image is shown on Figure 5.9)

5.2.3 How to estimate the position of the tumour cells in the descriptor curve?

The algorithm now is taking shape. The steps are:

1. The subimages belonging to the capsule are identified and tagged from the original image.

2. Back to the equalized image, the mean values of the entropy and the gray level differences for these capsule subimages are calculated.

3. Those subimages which do not belong to capsules but have entropy values higher than this mean entropy are classified as tumour cells.

4. Those subimages classified as cells but with entropy values greater than a threshold are further classified as tumour cells. The threshold after experimentation is

chosen to be the mean value for capsule minus 0.25 of the standard deviation. This inevitably mis-classified some normal cells as tumour cells. To rectify this mis-classification, those so classified tumour subimages will be re-classified as normal cells if the gray level difference is above a threshold (the mean gray level difference of the capsule plus 3.0 of standard deviation).

It is noted that in this part of the algorithm, the mean (average) values are used instead of the centroids. This is for computation speed, and as shown by the training set, the centroid is close to the mean value. But most important of all, we are dealing with the curve of the individual image concerned.

5.2.4 Refinements of the algorithm

The above algorithm generates two kinds of noises, one inherited from the nature of the ultrasonic images, and another from the arbitrary thresholds of the algorithm. Refinements to eliminate these noises are inevitable.

Structures outside the liver

As this research aims at automated screening of the ultrasonic image instead of the manually selected regions of interest as other researches described in Chapter 2, the algorithm needs to consider structures outside the liver but are included in the USG, e.g. part of the intestine and the kidney. The kidney, in special, gives features similar to a tumour although the human expert can easily differentiate it from the liver by its position (outside the liver). In the two-dimensional (B-mode) scan, it is below the posterior capsule. So any suspicious tumour subimages below the posterior capsule is a false signal and is to be excluded.

Fuzzy borders of the capsule

As discussed above, the subimages bordering the capsules are sources of inaccuracy due to the diffraction and reflection of sound waves from the capsule, and to some extent by the equalization of the image. The resultant haphazard intensities often lead to mis-classification of these subimages as tumour. A tumour subimage bordering immediately a capsule-subimage is most likely a mis-classification and is thus eliminated. Four-connectivity (those immediately above, below, to the left and to the right) is used to define "immediate border".

Algorithm-generated noises

The algorithm uses arbitrary thresholds which are set by experimentation rather than calculation. This inevitably introduces some noises. One method to exclude such noises is neighbourhood relationship discussed in Chapter 4 (Section 4.4). Again, the four-connectivity is a suitable criterion. The central subimage centred at a 3x3 filter is classified into the same cluster as the majority of its four immediately connected neighbour.

Even after this filtering, there remain some noises consequent to the pre-defined size of the subimage for statistical computation. This has been discussed in Chapter 1 (Section 1.4.2). The 10x10-pixel subimage represents an area of 0.22 cm^2 of the human liver. An aggregate of 10 subimages represents an area of 2.2 cm^2. If we set that a region is suspicious of tumour only if its area is at least 10 subimages, we then set the sensitivity of the algorithm to detect tumour size of at least 2.2 cm^2 (the best a human expert can achieve with the present scanners available).

5.3 *Results of the Algorithm*

The algorithm was tested on 15 USGs with carcinoma and eight USGs of normal liver (different from those used in Chapter 4), all randomly chosen. Significant suspicion means any aggregate of 10 subimages clustered as tumour cells. Table 5.2 shows the result:

USG	Number of suspicious subimages	Number of significant aggregates
C1	533	7
C2	182	2
C3	146	4
C4	85	4
C5	75	1
C6	111	4
C7	35	1
C8	16	0
C9	73	1
C10	50	1
C11	101	2
C12	25	1
C13	26	1
C14	137	4
C15	38	2
N1	11	0
N2	9	0
N3	4	0
N4*	42	0
N5	12	1
N6	10	0
N7	17	0
N8	17	0

Table 5.2 Number of subimages classified as tumour cells
(* N4 is an image containing a cyst)

The number of suspicious subimages is, as expected, much higher in the USGs with carcinoma. The number of significant aggregates reflect the size of the tumour if such one exists. In all the normal USGs except one (#N5, Table 5.2), there are no significant aggregates. In this aspect, the algorithm has a fairly high specificity. There is one carcinomatous USG (#C8, Figure 5.9) not successfully clustered. If we take that the other USGs are correctly clustered, the sensitivity of the algorithm is 93%.

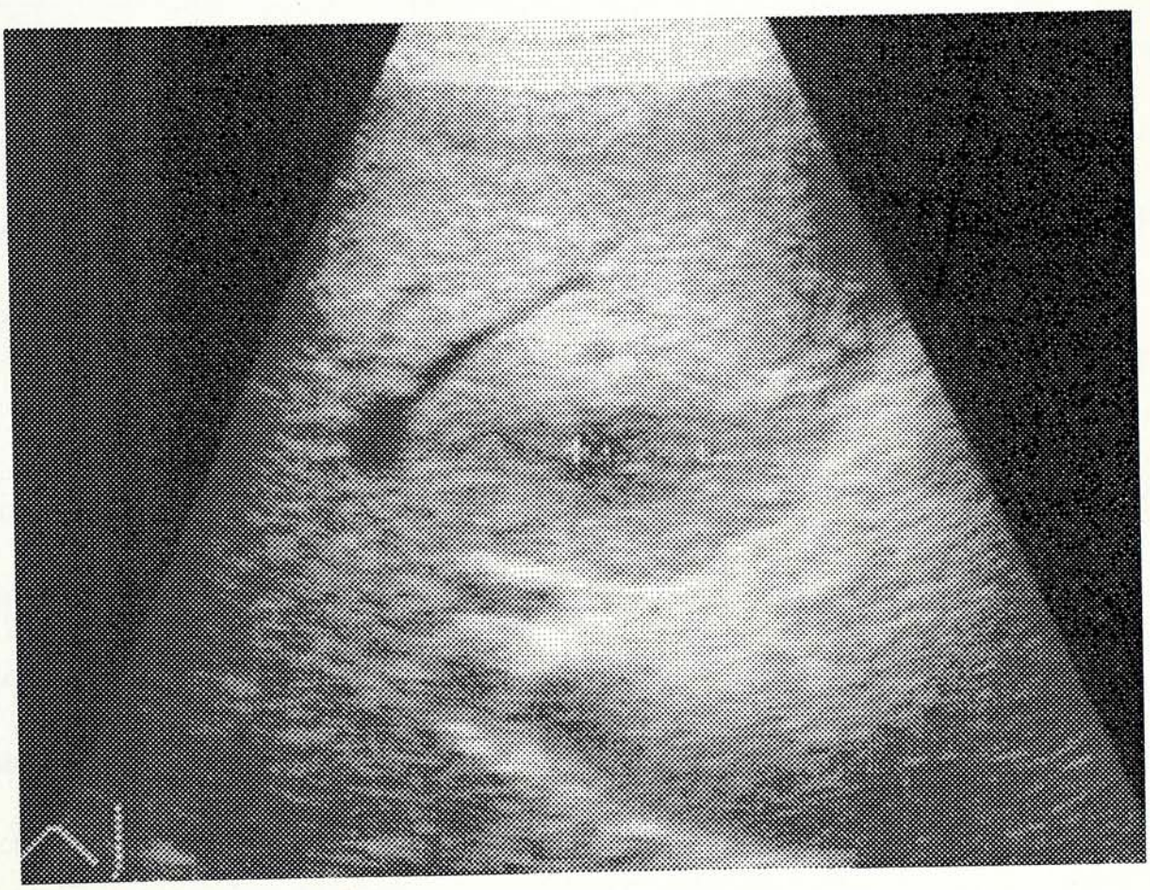


Figure 5.9 A small iso-echoic carcinoma (#C8, Table 5.2) not detected by the algorithm

Figure 5.9 shows the equalized image of the carcinomatous USG (#C8) that is not detected by the algorithm. The carcinoma (marked diagonally by two tiny crosses on the image) is small and iso-echoic. Such tumour is very difficult to detect even by human experts, especially on unequalized images.

How well will the algorithm work with hypo-echoic tumours? USG #C6 is an example. Figure 5.10 shows the equalized image and Figure 5.11 shows the corresponding result of clustering. Although the algorithm detects the presence of a tumour, it is not completely accurate in that it outlines the tissue immediately above the tumour. The tumour itself pushes and compresses the cells above causing a highly heterogeneous texture. The algorithm mistakes this heterogeneous texture and so hits the boundary.

This inaccuracy is acceptable in the light of the requirement of the objectives of screening and as a complementary tool to the human expert to alarm a warning signal for further scrutiny. In this aspect, the algorithm serves as a good though less than perfect guide.

Next, we shall see the algorithm's performance on hyper-echoic tumours. A liver USG showing a central carcinoma is shown in Figure 5.12, and the digitized result of the algorithm in Figure 5.13. There are four regions isolated to be suspicious of tumour. Their sizes are 1, 2, 4, and 63 subimages, the largest one (the only suspicious aggregate) being the tumour. The USG image actually shows a section of the left lobe of the liver with the upper pole of the kidney at the lower third of the image. The carcinoma is solitary, oval, small and hyper-echoic. Figure 5.13 demonstrates the excellent performance of the algorithm with nearly completely accurate localization of the tumour.

Examination of the results of other images tested also shows that the algorithm works very well with hyper-echoic carcinomas.

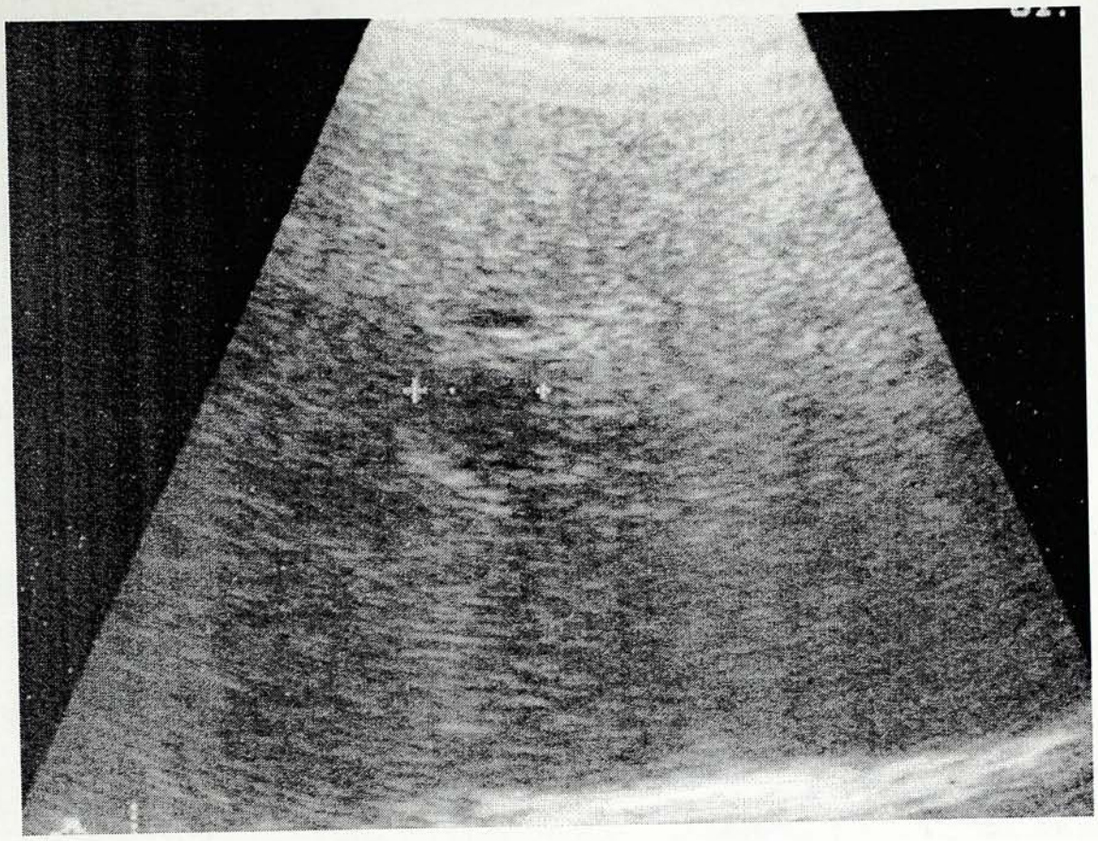


Figure 5.10 Equalised image of USG #C6 (Table 5.2) with a hypo-echoeic tumour (marked by crossess)

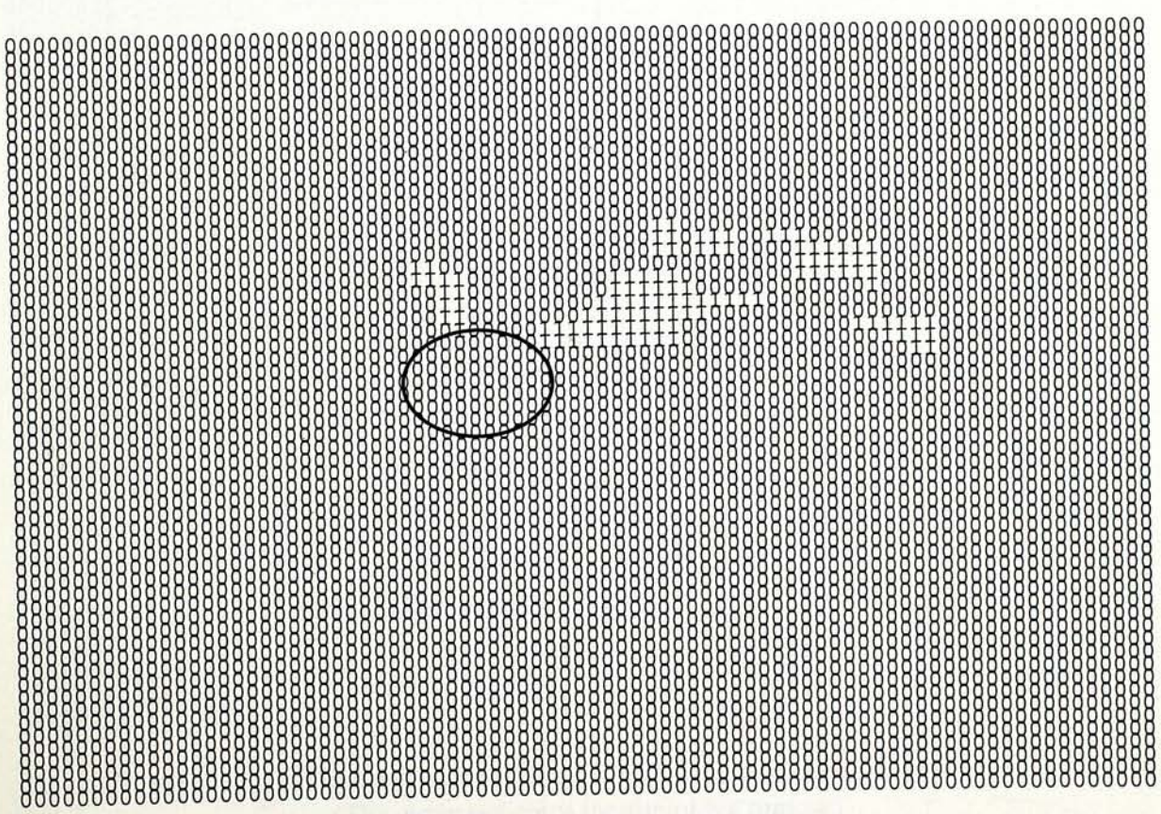


Figure 5.11 Result of clustering Figure 5.10 by the algorithm
(The circle indicates the site of the tumour.)

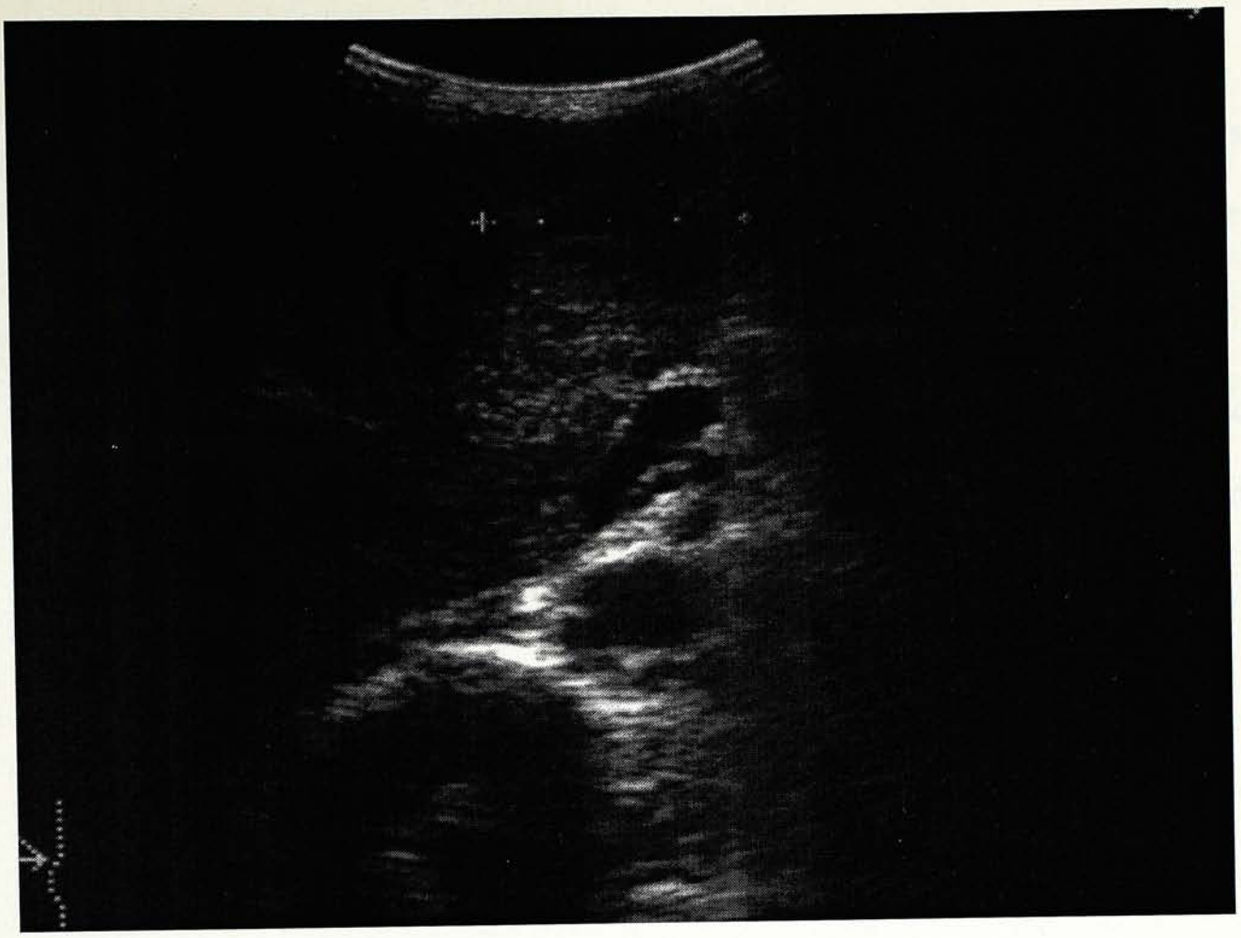


Figure 5.12 A liver USG showing a central hepatocellular carcinoma (

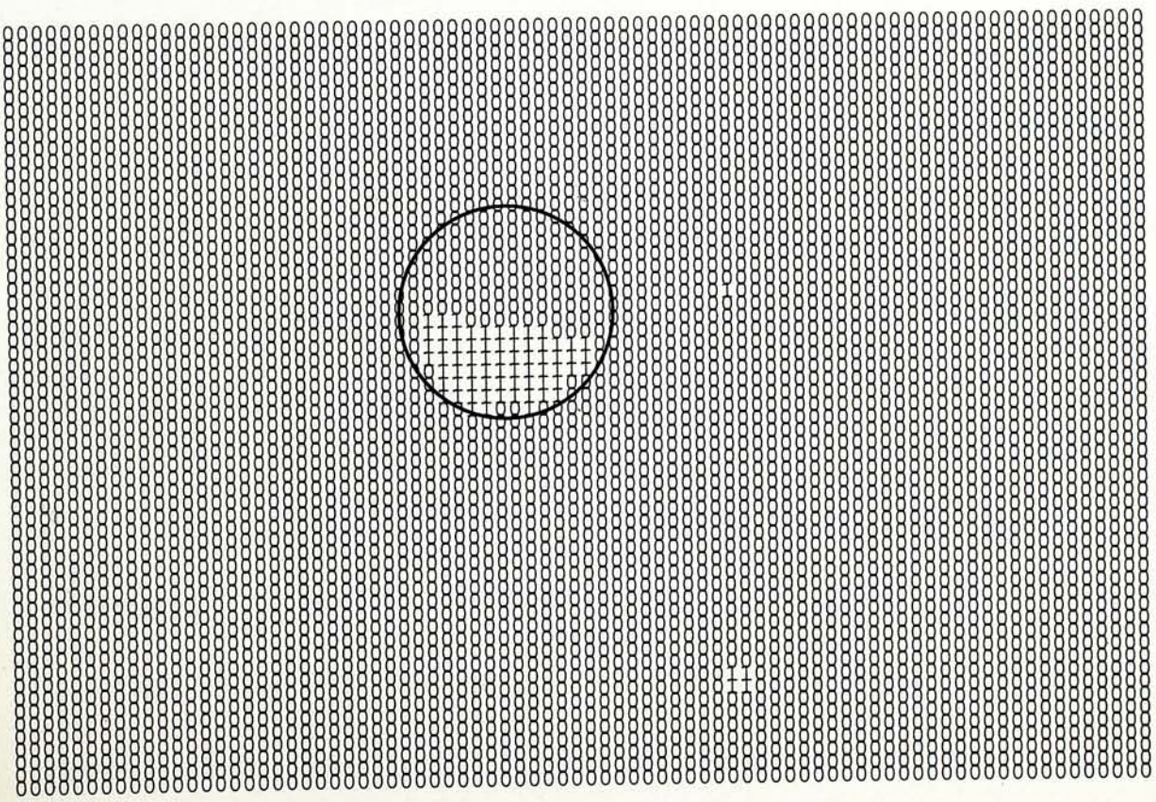


Figure 5.13 Digitized result of Figure 5.12 with suspicous tumour sites (
(The circle indicates the site of the tumour)

5.4 *Further Examples*

The following are a few images taken from those in Table 5.2. The images are printed in sizes 4x3 inches while the digitized results are printed in 80-column x 60-row matrices. This arrangement facilitates comparison.

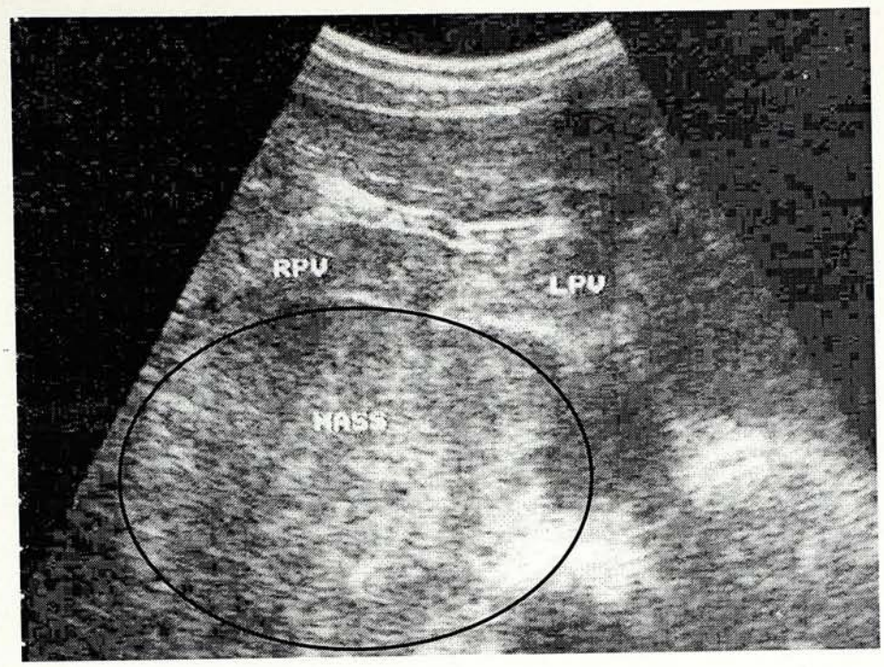
Figure 5.14 (#C11) shows a typical, large heterogeneous carcinoma below the right and left portal veins (RPV, LPV). Visually the tumour consists of regions of different textures and brightness. The algorithm identifies it as two suspicious regions.

Figure 5.15 (#C12) shows a large solitary tumour at the middle of the image. The algorithm identifies the site of the tumour though only part of it is isolated.

The tumour in Figure 5.16 (#C4) is outlined by a circular hypo-echoic zone. The algorithm identifies the upper and lower part of the circular tumour and the two suspicious regions form a semi-circular outline. There are two areas mistaken by the algorithm as suspicious regions. These are the artifacts bordering the posterior capsule.

In Figure 5.17 (#C3) there is a tumour at the upper left quadrant of the liver image. The lower right quadrant of the image is the kidney and the surrounding fat. The tumour is well isolated except the hypo-echoic part.

The image in Figure 5.18 (#C14) shows only one tumour before equalization of the gray level histogram (the dense one just below the mid-line). After equalization, two more tumours emerge (one at the left of the image, one is bud-like beside the first dense one). The dense tumour is isolated well by two suspicious regions. The left tumour is identified at its lower border.



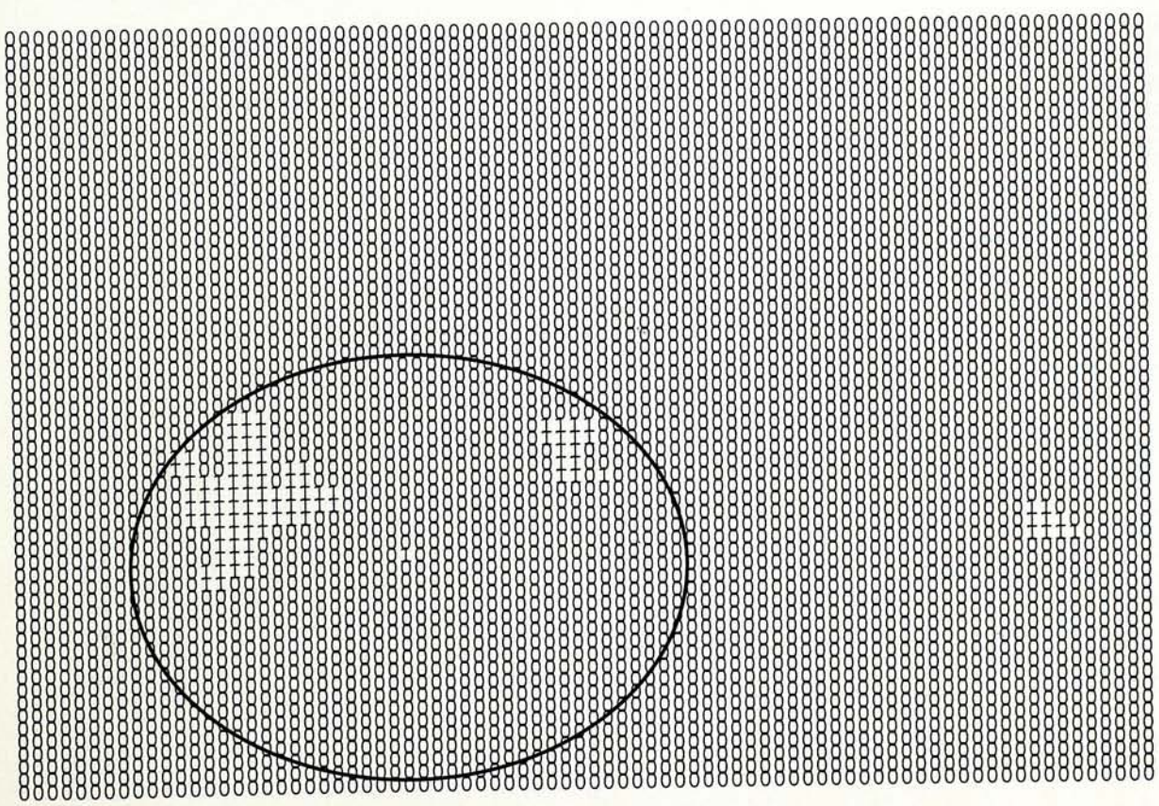


Figure 5.14 Carcinoma of #C11 and digitized result of clustering
(The circle indicates the site of the tumour)

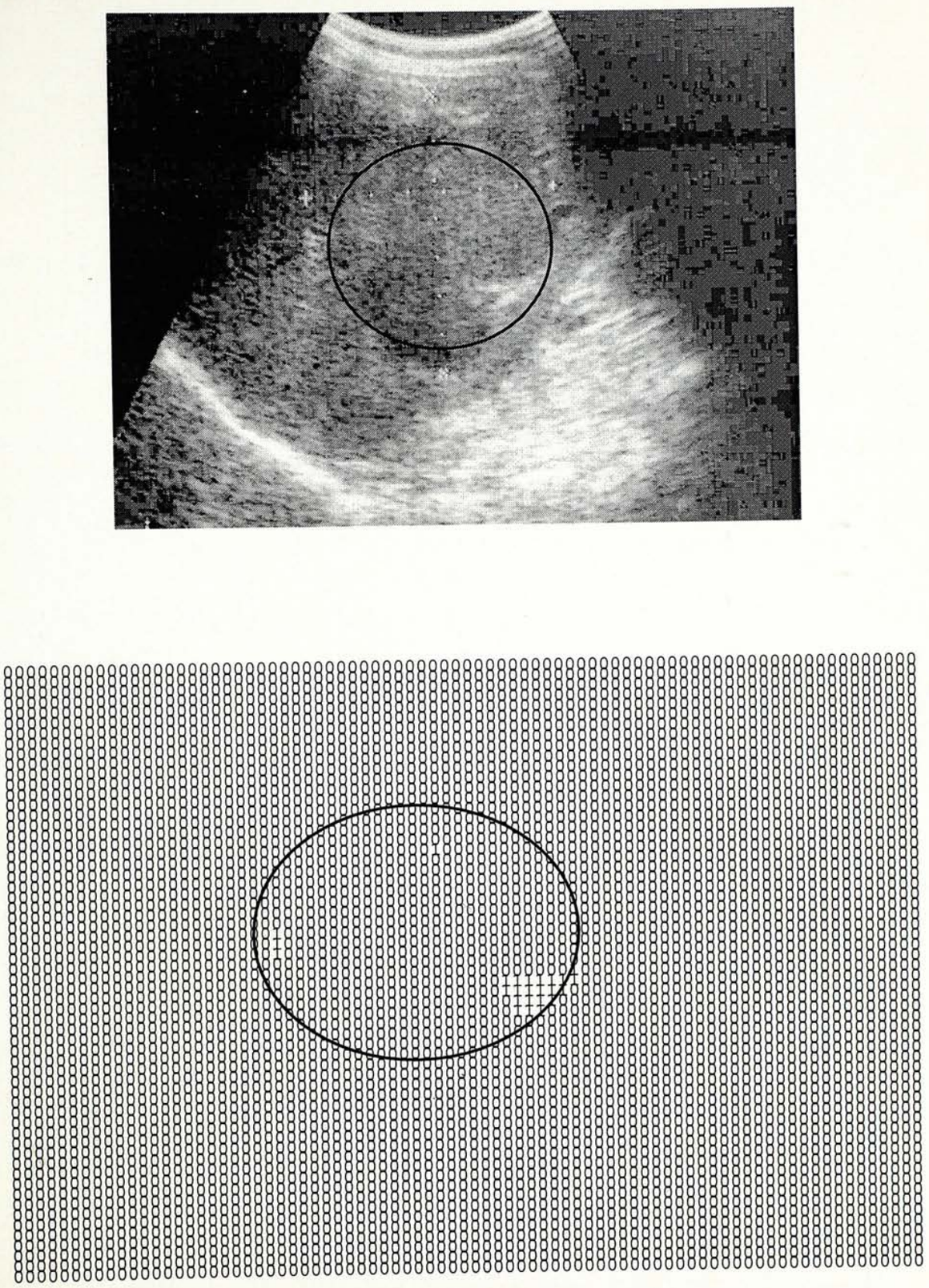
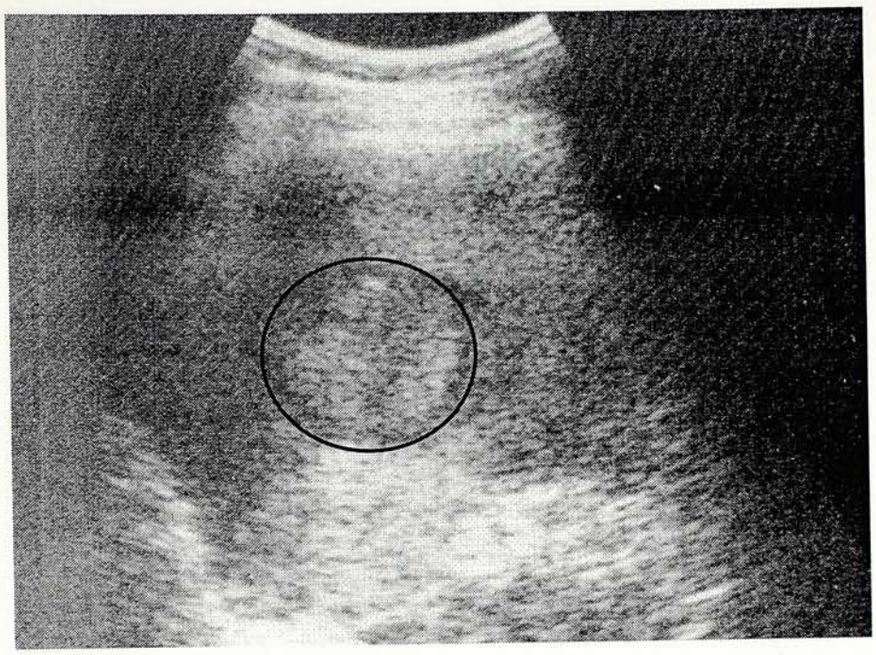


Figure 5.15 Carcinoma of #C12 and digitized result of clustering
(The circle indicates the site of the tumour.)



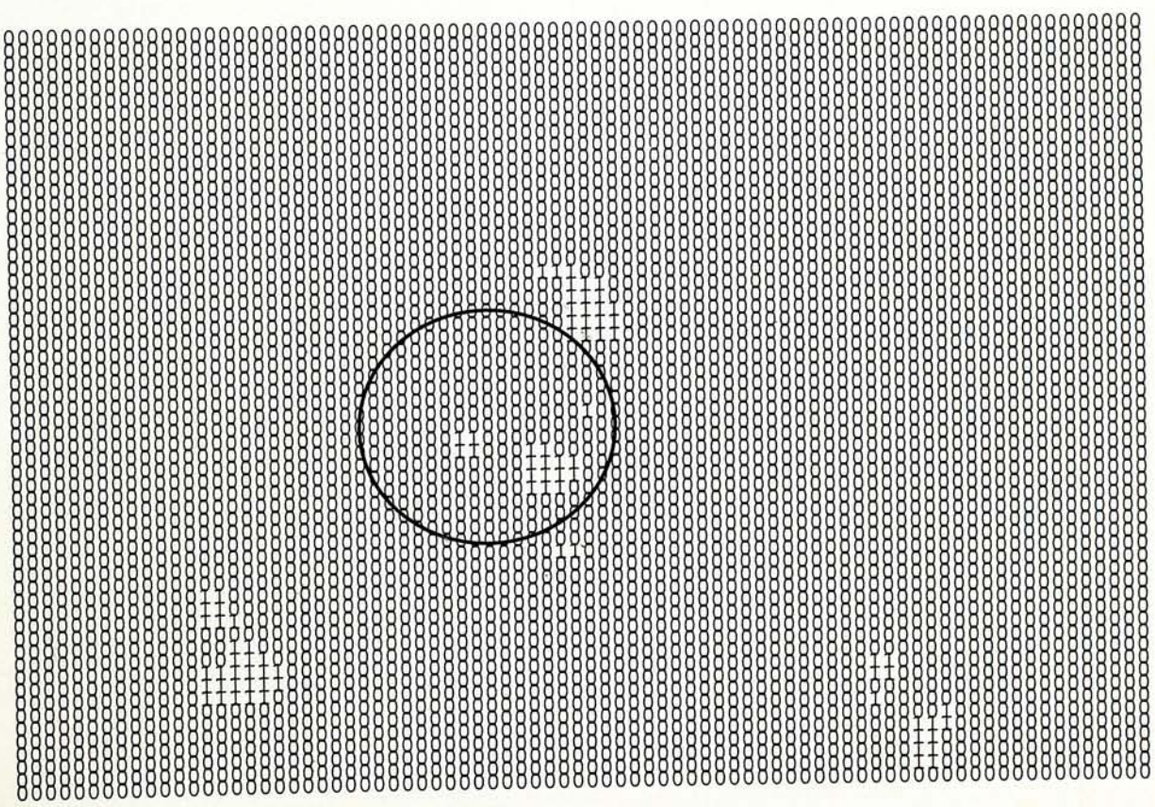


Figure 5.16 Carcinoma of #C4 and digitized result of clustering
(The circle indicates the site of the tumour)

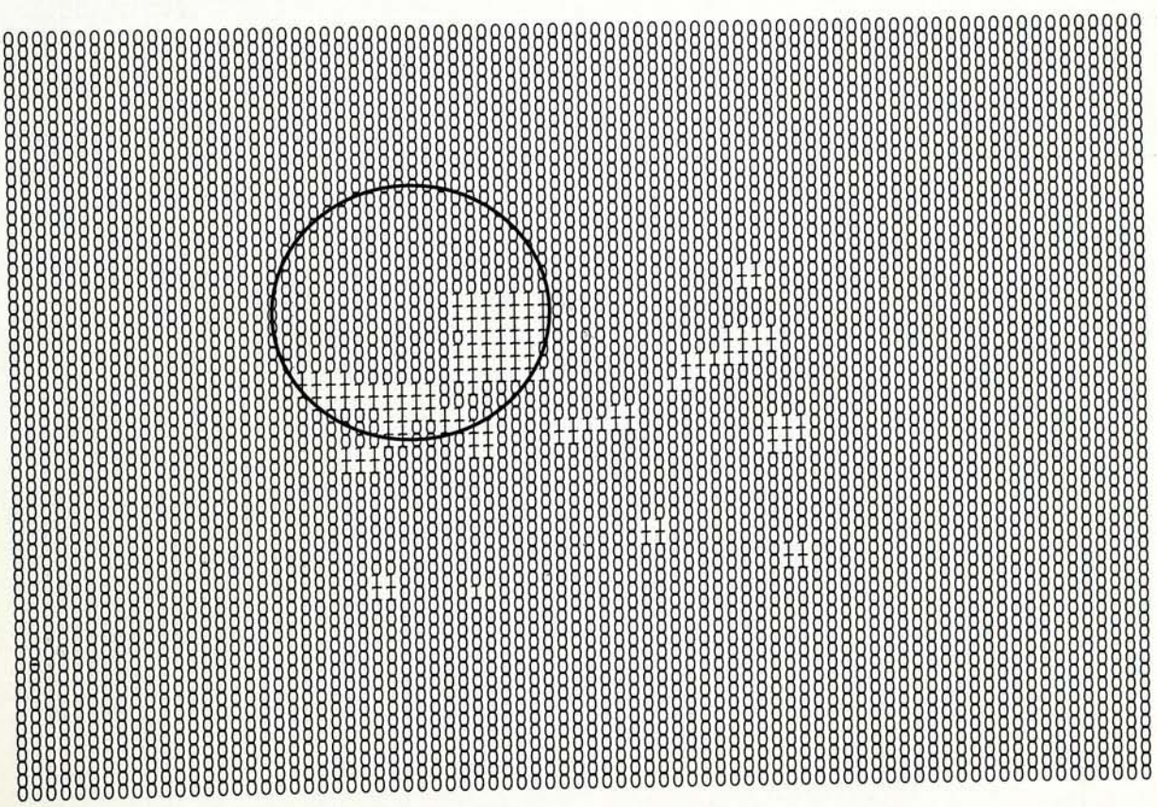


Figure 5.17 Carcinoma of #C3 and digitized result of clustering
(The circle indicates the site of the tumour)

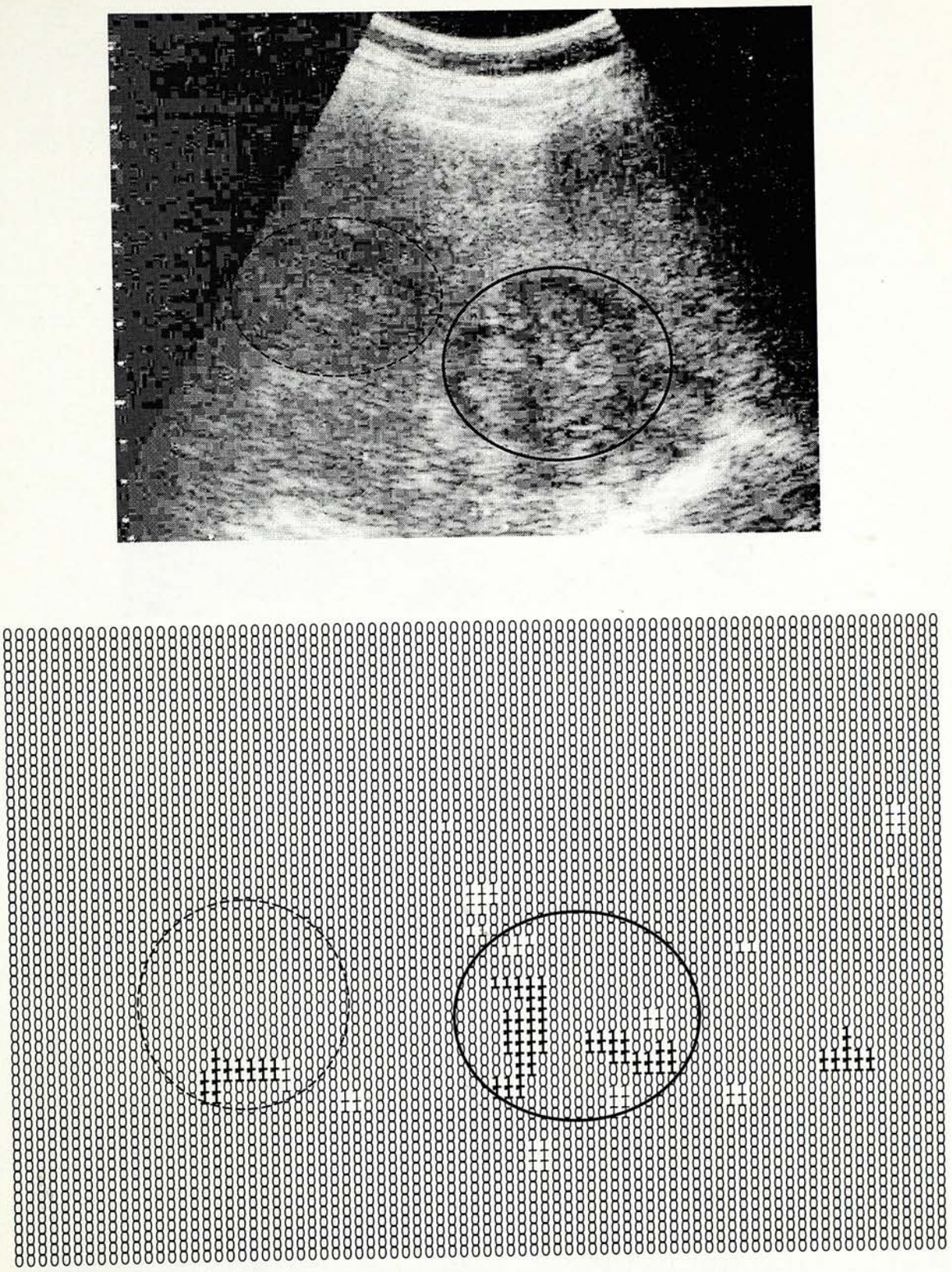


Figure 5.18 Carcinoma of #C14 and digitized result of clustering
(The solid circle indicates the site of tumour.
The broken circle indicates the suspicous tumour)

Fig. 5.19 shows a normal liver with a cyst near the anterior capsule. The algorithm does not confuse the enhancement below the cyst as suspicious tumour.

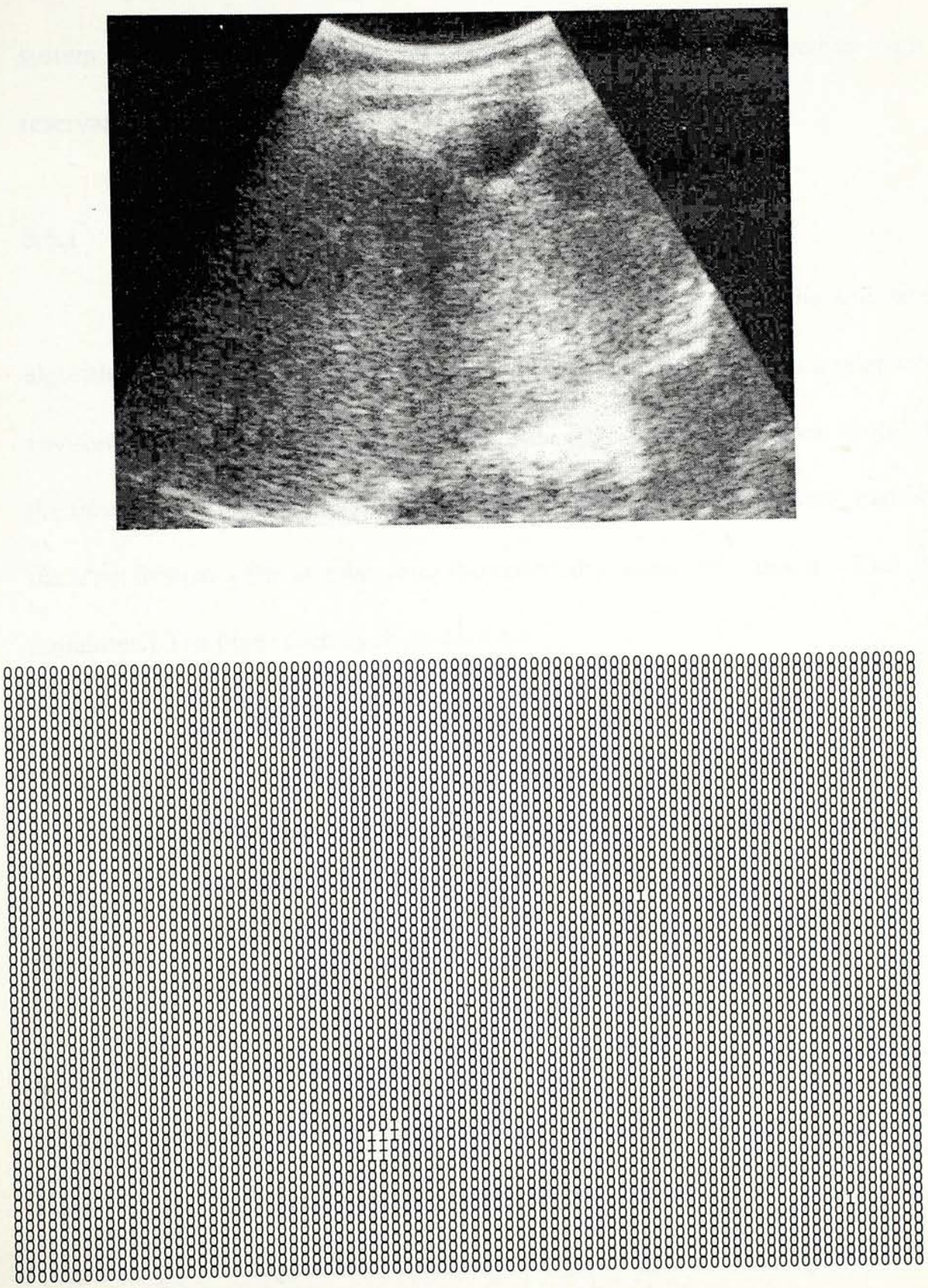


Figure 5.19 USG of a normal liver with a cyst and the digitized result of clustering (c.f. Figure 4.9)

5.5 *Evaluation of the Algorithm*

As stated in the Objectives of this thesis (Section 1.4), the constraints of the desired system for screening are speed, cost, ease of incorporation into present system and sensitivity. The present algorithm meets all these objectives with some reservations on the sensitivity.

5.5.1 Time required by the algorithm

The speed of an algorithm depends on the hardware and the software. This algorithm runs at a speed which is certainly acceptable. It is done in a relatively slow environment of a personal computer with Intel 80486 micro-processor at 66 Mhz and the time required is much less than the computation of a co-occurrence matrix under the Unix System. (The angular second-moment discussed in Section 3.3.2 takes about 2 minutes.) The time taken by the algorithm is:

digitization into raster image	25 sec
digitization into equalized data	29 sec
extraction of 5 statistical features	185 sec
isolation of supportive tissues	25 sec
tumour detection	3 sec
total	267 sec

Adjustment of the algorithm will further improve the speed. For detection of tumour subimages only, classification of the image into cells and background is not necessary and the algorithm may start at the isolation of the supportive tissues (capsule) by the edgeness operator. At its present form, the algorithm works in discrete steps as listed above and the feature extraction takes up most of the time required. The isolation of supportive tissue can be done simultaneously with the

digitization of the raster image data, and, similarly, the extraction of features can be done together with the equalization of the image data. The speed can thus be further shortened.

5.5.2 Sensitivity

On the testing set, the sensitivity is 93% which can be regarded as good. The only drawback is the precision in locating the exact site of the tumour. While the localization is accurate and exclusive with hyper-echoic tumours, the algorithm locates the immediate neighbour of hypo-echoic tumours and may miss an iso-echoic one. A plausible explanation is that the small training set contains more information about hyper-echoic than hypo-echoic tumours. While this outstandingly small training set demonstrates the efficiency of using the curve-shapes of statistical textural features, it puts a small trade-off to the effectiveness of the approach.

The algorithm has to deal with lots of noises as discussed in Section 5.2.4. Though it manages them well, it suffers from two decompensations. Firstly, it has to bring in steps to re-assess and re-adjust initially clustered subimages. This introduces more steps and computation time though the extra work is trivial. Secondly, it has to limit its sensitivity to significant aggregates of 10 suspicious subimages. In this aspect, it cannot outperform a human expert, though this is not the primary aim of the system. Further refinement on the elimination of noises will improve the sensitivity.

5.5.3 False positives and negatives

Closely related to the sensitivity is the presence of false positives and false negatives. By "false positives" the algorithm suggests the presence of tumour in

images which are normal, and by "false negatives" the reverse occurs. To assess these aspects, a very much larger testing sample (about thousands) than the present research is required. Such large sample size is outside the scope and practicability of the present thesis due to resources. (Chapter 6, Section 6.2, will further discuss this point. One of the objectives of the thesis is to prove the feasibility of such a system.)

Due to various noises in and the constraints of the system, it tends to err more on the false positives (see Section 1.4 "Objectives of thesis"). For the purposes of screening and assisting rather than replacing the human expert, this error (up to a certain limit) is relatively more acceptable than error on more false negatives.

The false positives can be minimized by fine tuning of the system. (1) Noises may be reduced by grabbing the images directly from the ultrasound scanner. (2) The thresholds on the gray level difference and the entropy of gray levels for the positions of tumour cells on the descriptor curves (Section 5.2.3) can be more meticulously adjusted. (3) The size of the aggregates (of suspicious tumour subimages) may further be adjusted for better accuracy and sensitivity.

CHAPTER 6

REVIEW and PROSPECTS

This thesis has so far described the background, motivation, and applications of the desired system to supplement human expert in the screening of hepatitis-B carriers for carcinoma of liver, using B-mode ultrasonography. It then designs the system by acquiring images from the hard copies of USGs of known diagnoses (by human experts), digitizing the images, and extracting statistical descriptors for a decision-tree algorithm. The algorithm first clusters the liver USG image into four macro-structures (anterior capsule, background, cells and posterior capsule) by taking the shortest Euclidean distance of selected statistical parameters (mean, gray level difference, entropy of the gray levels, roughness, and pixel difference) from the training set. The algorithm then detects regions of heterogeneity (gray level difference) and volatility of surface (entropy of gray levels) for suspicious regions of tumour cells. Examples demonstrate that the algorithm successfully detects images of liver carcinomas and identifies, with some constraints, the locations of carcinomas.

The hardwares are a personal computer and a flatbed scanner. The software is the algorithm written in C-language.

This Chapter is going to draw conclusions, have evaluations, and make recommendations for future research.

6.1 Conclusions

6.1.1. The objectives

This thesis has demonstrated that a fully automated system for the detection of liver carcinoma in USG is not only practical but also capable of satisfying the constraints of computer-aided screening: acceptable speed, inexpensive, sensitive, and ease of incorporation into existing systems. (Most new ultrasound scanners nowadays provide outlets to be connected to external computers.)

While the system satisfies these objectives, it is not intended to replace but to complement and assist human experts. Can this system stand alone in the screening process? In practice and contrary to the opinion that "with present available technology, no computerized system is capable of differentiating tissues without any guidance or interaction of the physician" [Schuster88], the answer is "yes". There is no human-machine interface throughout the system, i.e. it is fully automated. But theoretically and ethically, a human expert, rather a machine-expert, should be responsible for all medical decision.

6.1.2 Hypotheses

This thesis has proven by (a) extracting different statistical textural features from several liver USG images and (b) using these features to cluster successfully testing images the following hypotheses:

1. Different macro-structures of liver USG have characteristic statistical features.

2. The statistical features have special shapes when distribution curves are plotted against the macro-structures.

2a. The shape of a feature-structure plot reflects its nature of the feature, e.g. gray level difference reflects the differences between two neighbouring pixels.

2b. For each and every image, the structure plot for each statistical feature has not even the same shape but also closely similar absolute values (when the gray level histogram is equalized).

2c. The structure plots of each feature of different images can be adjusted to superimpose onto one and other.

3. From 2c above, an algorithm can cluster liver USG images effectively and efficiently.

6.1.3. Statistical features

The thesis has shown:

1. Statistical textural features characterize the different macro-structures of the liver USG.
2. The most single discriminating statistical feature is the mean gray level of the image.
3. Second-order statistics are important and indispensable for the clustering of macro-structures perceived by human vision.
4. Among the second-order statistics, the gray level difference and entropy are more consistent and useful in discriminating the macro-structures.

5. Three novel second-order statistics (the edgeness, pixel difference, and roughness) are as consistent and useful as the gray level difference and entropy.

6. The fractal dimension and run length percentage are not as consistent nor useful as the other features.

6.2 Evaluation

This thesis has illustrated the fundamental principles and demonstrated the feasibility of the algorithm. Though the algorithm satisfies the objectives of the thesis and proves the hypotheses, it inherits several deficiencies. The ideal test is one that has high sensitivity (detecting all those with the condition) and high specificity (excluding all those without the condition). No single test has 100% sensitivity and specificity. All tests, in any scientific discipline, suffer in various degrees from false positives as well as false negatives. The system in this thesis is no exemption (Table 5.2). But to test these properties fully, a large sample size of patients no less than hundreds and beyond the scope of a single centre is required. For 1993 the Death Register in Hong Kong recorded 1161 deaths of carcinoma of liver [HongKong94]. Hence coordinated co-operation from several hospitals for a period of about one year is required for stringent values of sensitivity and specificity (this will be discussed in detail in a later section). Suffice it to say at this point that the results from this algorithm are valid and reliable, as shown in Section 5.3.

6.2.1 Noises

Section 5.2.4 has discussed the problem of noises from this algorithm. Noises may be regarded as false positives and are a by-product of attempting for high

sensitivity (trying to include all the positives is likely to recruit those near-positives). There are two aspects on noises of this algorithm that need further comment.

(a) The determination of subimages from the entropy and gray level difference.

The thresholds for the tumour subimages are estimated by trial by error. This estimation lacks mathematical precision. A more precise threshold might well decrease lots of noises.

(b) The "suspicious subimages".

The magic figure of 10 is used to assign the size of aggregates of subimages to be suspicious of tumour cells. The basis of this size is the smallest size of a real tumour claimed to be detected by human experts. This basis is not optimal. The computer may outperform a human. The best method to determine this size (if noises cannot be reduced to near-nil) is to try a large sample of normal liver images. The size that gives the least number of false negatives (i.e. the highest specificity) should be the optimal choice.

6.2.2 Statistical features

There is no doubt for the value of statistical textural features in segmentation and clustering. The third-order statistic run length does not play any role while the first-order statistics (the mean and the entropy of the gray levels) play very important role in clustering the images in this thesis. The experiments by Layer et al with laboratory rats found that the mean gray level was the most discriminating feature in differentiating normal and cirrhotic liver cells [Layer90]. But their experiments were over-simplified in the aspect of image-segmentation as they took only regions of

interest and hence avoided the fuzzy boundaries and feature overlaps between different macro-structures. This thesis also finds that the mean gray level is the most consistent and important feature for separating different macro-structures. But other features (mainly the second-order statistics) are required to deal with the fuzzy boundaries.

Edgeness

The usefulness of this feature as compared with the Sobel gradient has been demonstrated in Section 5.2.2. The only uncertainty is this feature's robustness of the threshold. For the images taken for this thesis, the threshold of 10 gray levels is very effective. But would this same threshold be as effective for other images from other machines? Would this threshold be USG-dependent, and then in-turn operator- and machine-dependent?

Pixel Difference

The pixel difference takes the gray level difference at the direction $\delta(\theta,r)$ with r=1 and θ=0° and 90°. It is the combination of the gray level difference and the Robert's gradient. It takes just one δ value. If more δ values are taken, the pixel difference acts like a derivative of the co-occurrence matrix. It is equivalent to the run difference matrix proposed by Kim [Kim91].

The pixel difference may be a more powerful descriptor if at least one more δ value is taken. In the latter case, there is a ratio (relative to different δ values), e.g.

$$PD_{ij} \text{ ratio} = \text{pixel difference at } \delta_i \text{ / pixel difference at } \delta_j$$

Albregtsen investigated similar ratio (gray level run length at two different quantization widths) and found satisfactory results in discriminating between normal and malignant cell nuclei [Albregtsen92]. It would be interesting to have further experiments on this statistical feature.

Roughness

The roughness is theoretically a better parameter than the standard deviation in measuring the heterogeneity of a region or a pixel relative to a defined neighbourhood. The standard deviation measures the dispersion of a pixel from the mean value of its neighbour but the roughness measures the heterogeneity of a surface or a pixel relative to the surface centre. Taking this idea further, if one takes a pixel as the centre of a region and measures the roughness of this pixel in different sizes of neighbourhood, e.g. 3x3, 4x4, 5x5, we may have a parameter analogous to the fractal dimension of that pixel (or a point) on that surface.

The roughness may be another method for the computation of the fractal dimension. It is yet uncertain about the possible applications of the fractal dimension of a point instead of a surface. Would that be a good edge-detector? Again, it would be interesting too to pursuit this feature further.

6.2.3 Methodology

Using non-overlapping subimages for clustering has the advantages of computational speed but at the same time inherits two disadvantages. First there is no algorithm to determine the best suitable size of the subimage. Heuristic methods like trial by error are used in this thesis. Second, some subimages would inevitably stride over different regions and mis-classification ("noise") occurs along boundaries. This

algorithm arbitrarily classifies these subimages into "capsules" in case they have been clustered into "tumour cells". But there is no strategy to handle these boundary subimages for images of normal liver USGs.

Because this thesis is an experiment to build a new system and there is no funding for or provision of hardwares, the algorithm builds on images scanned from hard copies, instead of direct grabbing from the ultrasound scanners. This approach was thought to involve one extra step from the scanner screen to the personal computer. However it introduced one more source of variation in the experimental images. In the initial stage of the research, the setting of the scanning parameters (e.g. the gamma value, the brightness, the contrast) was not uniform and inconsistencies were observed among the images. The importance of this source of variation was later realized. The images were then scanned with a uniform setting of parameters that gave same appearance (judged by human eyes) as the original hard copies of USGs. The same scanner and the same procedures were used throughout the experiments. This approach minimizes the possible variations in images but is still short of the ideal of direct capture of images from the ultrasound scanning. Even in the later approach, the image-capture must be standardized.

What are the inconsistencies and effects from the variation in scanning hard copies of USGs? To answer this question, images of different scanning settings were studied. The shapes of the statistical feature plots remain the same but the absolute values change so that the vertical shift among different images fluctuates widely. The run length percentages are most inconsistent — a fact that suggests why this feature is not useful in the algorithm. The ultrasonographer adjusted his images while scanning the patients and this is not corrected by gray level histogram equalization.

The unnecessary variation may potentially affect this algorithm in giving variable centroids from the training set and hence different clustering results in the algorithm.

6.3 Future Work and Research

As pointed out previously, this thesis aims to prove the plausibility, feasibility and viability of a system. In the course of the proof, there are hypotheses and design of texture statistics. Obviously, two aspects need further pursuit. One is the practical implementation of the system and the another one is the theoretical exploration of the novel second-order statistics.

6.3.1 Implementation and further development of the system

The need of a large sample of USGs to eliminate noises is discussed in Section 6.2.1. A wide-scaled and coordinated program is required to collect a large sample of USGs of liver carcinoma (not for the training set, which can be as small as less than ten, but to test the sensitivity and specificity of the system). For the training set, more samples of the uncommon iso-echoic and hypo-echoic tumours should be included.

The methodology will be refined. The ultrasound scanner is to be directly connected to the computer which captures the image from the scanner screen.

At present, USGs are taken totally manually. The operator searches the whole organ by moving the ultrasound inducer along the body surface. This can be replaced by the machine moving the inducer systematically as if cutting the organ into slices, each slice being one image. This approach is similar to the computer aided tomographic scan (CAT scan). Each image is analyzed for suspicious regions of

tumour cells. Suspicious images are then tagged and suspicious regions outlined. Full automation is then achieved.

It is mandatory to have funding for this part of the research and implementation.

6.3.2 Future research of the system

Results from the last Chapter show that some tumour sites (especially those hypo-echoic tumours) have their immediate neighbour, instead of themselves, identified as suspicious regions. This is because the present algorithm takes into consideration the local statistical textural features. The structural features like the shape, area, perimeter are not considered. Some characteristic features of the liver tumour are not shown though the presence of such a tumour is noted. The best example is the characteristic "bulls' eye" in which the tumour consists of two parts: one hypo-echoic circle embracing an iso-echoic or hyper-echoic circle (much like the bull's eyeball). While not very common, this appearance is the typical tumour feature by which the human expert has no doubt about the diagnosis. To include this and similar features and to enhance better delineation of the tumour regions, approaches in addition to local statistical features are required. The trade-off is of course the computation speed and/or total cost.

The next approach for refinement of the system would be the development of an expert system which is briefly discussed in Chapter 2. Structural information like shape, area, will not give more intelligent interpretation to differentiate tumour from, for example, cyst or haemangioma (collection of abnormal blood vessels). An expert system is close to human logic by considering combinations and relationships of

features. Moreover, there is no way to identify ultrasound artifacts like enhancement and shadowing apart by expert systems. An expert system is probably the final version of automated screening though it may not be a real-time computer-aided screening due to the much more computation time required. Certainly, artificial intelligence with roots on statistical texture analysis will be the path for further research.

6.3.3 Fuzzy algorithm

It is only logical to think that fuzzy algorithms may solve problems of fuzzy regions contained in USGs. That this thesis finds fuzzy algorithms unable to solve the problem does not suggest that fuzzy logic is not applicable in USGs. Further studies with fuzzy algorithms are required. Intuitively, I expect the combination of fuzzy algorithm and artificial intelligence is the ultimate solution for fully automated detection and localization of tumour cells in USGs with the most important constraint of computation speed.

6.3.4 Further work on statistical texture features

As discussed in Section 6.2.2 the novel statistical features devised in this thesis have not been fully studied. They are used by the algorithm to meet its objectives but their potential in clustering or segmentation has not been fully explored. It is difficult to say if they are useful in other applications of image analysis.

6.3.5 The commercial potential of the system

I have discussed in Section 1.2 the overwhelming demand for ultrasonograms of liver in Hong Kong. To assist detection of liver carcinoma, the present algorithm and system require only little modification of existing machines. Moreover, the

principle can be extended and applied to carcinoma of other internal organs like the uterus requiring ultrasound investigations. After further research and refinement, the present system can be developed into a commercial package which serves as an add-on equipment to the ultrasound machines.

6.4 Final Conclusion

1. This thesis demonstrates how an algorithm can detect regions suspicious of cancer cells on images of liver USGs. The algorithm is fully automated, relatively fast, sensitive, inexpensive, and easily incorporated into existing ultrasound scanners.

2. Statistical texture features of the liver USGs have consistent and stable curve-shapes among different macro-structures. For the same feature, different images have structure-plots very close to one and each other if they are derived from equalized (gray level histogram) images.

3. The shapes of these feature curves enable effective and efficient clustering of the images as well as automated detection of regions suspicious of liver carcinoma.

4. The mean gray level is the most discriminating feature in clustering the liver USGs. The gray level difference and the entropy are also very useful.

5. Three second-order statistical features devoid of the computational complexities of the co-occurrence matrix are devised for the algorithm. One of these, the "edgeness" is more effective than the Sobel gradient in clustering subimages containing edges.

APPENDICES

Appendix A: Program Listings

The image processing and analysis in this thesis is done with programs written in Borland C and run on an IBM-compatible PC (486 processor and 16 MB RAM) under Microsoft DOS. Due to the limited available memory size, the programming technique is slightly different from that developed to be run at workstations; some steps could be shortened or truncated otherwise. The source codes for the essential algorithms are:

Listing 1: pcx.c

Listing 2: feature.c

Listing 3: detect.c

Listing 4: centroid.c

Other algorithms developed in the experiments, e.g. co-occurrence matrix, fuzzy logic classifications, region-growing by blob colouring, will not be listed as they do not give better results than the present algorithms, or finally not used.

🗁 Listing 1: pcx.c

This algorithm reads a 8-bit 256-gray level ".pcx" (Paint-Brush™) and digitzse the gray level into an ASCII file for further processing or analysis. It gives the operator the option to digitalise the image in its original form or with equalised histogram. It also displays the image (original or equalised) on the monitor screen in the presence of a SVGA display-card and SVGA driver (SVGA256.BGI developed by Jordan Hargraphix Software, Copyright 1990, available as shareware).

```
#include <stdio.h>
#include <dos.h>
#include <graphics.h>
#include <stdlib.h>
#include <string.h>

#define DISPLAY_MAX_X  800
#define DISPLAY_MAX_Y  600
#define MODE           0x00030

int Equalized;
int ReMap[256];
float Frequency[256], gray[256], equalizedGray[256];
double sqrt (double x);
unsigned char *data;
typedef unsigned char DacPalette[256][3];

typedef struct
{
   int count;
   char value[256*3];
} PALETTE;

PALETTE palette;

static struct PCXHEADER
{
   char manufacturer;  /* always 0x0a */
   char version;
   char encoding;
   char bitsPerPixel;    /* 256 color = 8 */
   int xMin;
   int yMin;
   int xMax;
   int yMax;
   int hResolution;
   int vResolution;
   char palette[48];
   char videoMode;     /* not used */
   char colorPlanes;     /* 256 color =1 */
   int bytesPerLine;     /* must be even no. */
   int paletteType;      /* 1=color, 2=grayscale */
   int shResolution;
   int svResolution;
   char filler[54];
} pcxheader;

/* Setvgapalette sets the entire 256 color palette */
/* PalBuf contains RGB values for all 256 colors   */
/* R,G,B values range from 0 to 63                 */
void setvgapalette(DacPalette *PalBuf)
{
/* These are the currently supported modes */
#define  VGA320x200    0   /* Standard VGA */
#define  SVGA640x400   1   /* 640x400x256 Svga */
#define  SVGA640x480   2   /* 640x480x256 Svga */
#define  SVGA800x600   3   /* 800x600x256 Svga */
#define  SVGA1024x768  4   /* 1024x768x256 Svga */

  struct REGPACK reg;

  reg.r_ax = 0x1012;
```

```
 reg.r_bx = 0;
 reg.r_cx = 256;
 reg.r_es = FP_SEG(PalBuf);
 reg.r_dx = FP_OFF(PalBuf);
 intr(0x10,&reg);
}

int vgamode()
{
   int mode;

   printf ("Choose a vga mode:-\n");
   printf ("VGA320x200   (Standard VGA)          --> 0 \n");
   printf ("SVGA640x400  (640x400x256 Svga)   --> 1 \n");
   printf ("SVGA640x480  (640x480x256 Svga)   --> 2 \n");
   printf ("SVGA800x600  (800x600x256 Svga)   --> 3 \n");
   printf ("SVGA1024x768 (1024x768x256 Svga) --> 4 \n");
   scanf  ("%d", &mode);
   return mode;
}

void vga256_set_palette(void)
{
   union REGS regs;
   struct SREGS sregs;
   int i;

   for (i=0;i<768;i++) palette.value[i]=(palette.value[i]>>2) & 0x3f;

   regs.x.ax=0x1012;
   regs.x.bx=0;
   regs.x.cx=256;
   regs.x.dx=FP_OFF(palette.value);
   sregs.es=FP_SEG(palette.value);
   int86x(0x10,&regs,&regs,&sregs);
}

int vga256_set_mode(void)
{
   union REGS regs;
   regs.x.ax=0x4f02;
   regs.x.bx=0x100;
   int86(0x10,&regs,&regs);
   return 0;
}

void vga256_draw_line(int y, int width ,char *data)
{
 int i;
 setcolor(256);
 for(i=0; i<width; I++) putpixel(i,y,(int)data[i]);
}

int get_pcxheader(FILE *f)
{
 if ((fread(&pcxheader,sizeof(pcxheader),1,f))==0)
     return -1;
 else
     return 0;
}

void display_pcxheader()
{
   printf("***************************************\n");
   printf("Header information of input PCX file.\n");
   printf("***************************************\n");
   printf("manufacturer= %d\n",pcxheader.manufacturer);
   switch(pcxheader.version)
   {
     case 0: printf("version= 2.5\n");
             break;
     case 2: printf("version= 2.8 with palette\n");
             break;
     case 3: printf("version= 2.8\n without palette");
             break;
     case 5: printf("version= 3.0\n");
```

```
            break;
 };
 printf("encoding= %d \n",pcxheader.encoding);
 printf("bitsPerPixel= %d\n",pcxheader.bitsPerPixel);
 printf("xMin= %d\n",pcxheader.xMin);
 printf("yMin= %d\n",pcxheader.yMin);
 printf("xMax= %d\n",pcxheader.xMax);
 printf("yMax= %d\n",pcxheader.yMax);
 printf("hResolution = %d\n",pcxheader.hResolution);
 printf("vResolution = %d\n",pcxheader.vResolution);
 printf("palette[48] = %s\n",pcxheader.palette);
 printf("videoMode = %d \n",pcxheader.videoMode);
 printf("colorPlanes = %d\n",pcxheader.colorPlanes);
 printf("bytesPerLine = %d \n",pcxheader.bytesPerLine);
 switch(pcxheader.paletteType)
 {
  case 1: printf("paletteType = Color\n");
          break;
  case 2: printf("paletteType = Grayscale\n");
          break;
 };

 printf("shResolution = %d\n",pcxheader.shResolution);
 printf("svResolution = %d\n",pcxheader.svResolution);
 printf("filler[54] = something\n",pcxheader.filler);
 if (Equalized) printf("Equalized = Yes\n");
 else printf("Equalized = No\n");
}

int get_pcx_palette(FILE *f)
{
 fseek(f,-769L, SEEK_END);
 if (fgetc(f)!=0x0c)
    return(-1);
 fread(palette.value,1,768,f);
    return 0;
}

int get_pcx_line(int n,char *data,FILE *f)
{
 int value,count;

 while ( n >0)
 {
 if ((value=fgetc(f))==EOF) return(-1);
 value &= 0xFF;
 if ((value & 0xC0)==0xC0)
 {
  /* two first two bit set to 1 */
  count=value & 0x3F;
  if ((value=fgetc(f))==EOF) return(-1);
  if (count>n) count=n;
  while (count)
  {
   if (Equalized) {
      *data++ = ReMap[value];
      gray[value]++;
      equalizedGray[ReMap[value]]++;
   }
   else {
      *data++ = value;
       gray[value]++;
   }
   n--;
   count--;
  }
 }
 else   /* normal data */
 {
  if (Equalized) {
   *data++ = ReMap[value];
   gray[value]++;
   equalizedGray[ReMap[value]]++;
  }
  else {
   *data++ = value;
```

```
 gray[value]++;
 }
 n--;
 }
}; /* end of while loop */
return 0;
}

int get_pcx_line1(int n,char *data,FILE *f)
{
 int value,count;

 while ( n >0)
 {
 if ((value=fgetc(f))==EOF) return(-1);
 value &= 0xFF;
 if ((value & 0xC0)==0xC0)
 {
 count=value & 0x3F;
 if ((value=fgetc(f))==EOF) return(-1);
 if (count>n) count=n;
 while (count)
 {
 *data++ = value;
 n--;
 count--;
 }
 }
 else   /* normal data */
 {
 *data++ = value;
 n--;
 }
 }; /* end of while loop */
 return 0;
 }

 void out_data(FILE *out,int width)
 {
    int i;
    for (i=0;i<width; I++)  fprintf(out,"%d ",(int)(data[i]));
 }

 int equalize (FILE *f)
 {
 register j;
 int row;
 int i=0;
 float Pixel_no=0.0, cum=0.0;

 for (j=0; j<256; j++)
 {
 Frequency[j]=0.0;
 ReMap[j]=0;
 }

 fseek(f,128L,SEEK_SET);
 row= pcxheader.yMax-pcxheader.yMin+1;
 while (row > 0)
 {
 get_pcx_line1(pcxheader.bytesPerLine*pcxheader.colorPlanes,data,f);
 for (j=0;j<pcxheader.bytesPerLine*pcxheader.colorPlanes; j++)
 {
 Frequency[data[j]]++;
 Pixel_no++;
 }
 row--;
 i++;
 if (i>=DISPLAY_MAX_Y) break;
 };
 for(j=0;j<256;j++)
 {
 cum+=Frequency[j];
 ReMap[j] = (int) (cum * 255.0/Pixel_no );
 }
 return 0;
```

```
}

int get_data(FILE *f,FILE *out)
{
  int row;
  int i=0;
  int temp;

  fseek(f,128L,SEEK_SET);
  temp=DISPLAY_MAX_X-1;
  row= pcxheader.yMax-pcxheader.yMin+1;
  while (row > 0)
  {
   get_pcx_line(pcxheader.bytesPerLine*pcxheader.colorPlanes,data,f);
   vga256_draw_line(i,temp,data);
   out_data(out,pcxheader.bytesPerLine*pcxheader.colorPlanes);
   row--;
   i++;
   if (i>=DISPLAY_MAX_Y) break;
  };
  return 0;
}

int huge mode()
{
 return 3;
}

void statistics(float gray[])
{
 register int i;
 double sum_x=0.0, sum_fx=0.0, sum_fx2=0.0, mean=0.0, deviation=0.0;

 for (i=0; i<256; i++)
 {
 sum_x += (float) gray[i];
 sum_fx += (float) i*gray[i];
 sum_fx2 += (float) i*i*gray[i];
 }
 mean = sum_fx/sum_x;
 deviation = (double) sqrt( (double) (((sum_fx2)/sum_x) - (mean*mean)) );

 printf ("sum_fx2 = %f\n", sum_fx2);
 printf ("total = %.f\n", sum_x);
 printf ("Mean = %f\nDeviation = %f\n\n", mean, deviation);
 }

 void main()
 {
    FILE *pcxfile, *datafile, *headerfile;
    int gd=DETECT, gm;
    char pcxname[40], dataname[40], headername[40];

    for (gm=0; gm<256; gm++) gray[gm]=0.0;

    printf ("Name of pcx file: ");
    gets(pcxname);
    if ((pcxfile=fopen(pcxname,"rb"))==NULL) {
              printf ("Open %s error! \n", pcxname);
              exit(1);
    }
    printf ("Name of data file to be saved: ");
    gets(dataname);
    if ((datafile=fopen(dataname,"wt"))==NULL) {
              printf ("Open %s error! \n", dataname);
              exit(1);
    }
    printf ("Name of header file to be saved: ");
    gets(headername);
    if ((headerfile=fopen(headername,"wb"))==NULL) {
              printf ("Open %s error! \n", headername);
              exit(1);
    }

    printf ("Equalise data? \n");
    printf ("0. NOT equalise\n");
```

```
printf ("1. Equalise\n");
scanf  ("%d", &Equalized);

if ( get_pcxheader(pcxfile)==-1) printf("Error in header!\n");
fwrite (&pcxheader, sizeof(pcxheader), 1, headerfile);
fclose(headerfile);

palette.count=256;
if ( get_pcx_palette(pcxfile) == -1)
        printf("Error in palette!\n");
display_pcxheader();
fprintf (datafile, "%d %d ", pcxheader.xMax+1, pcxheader.yMax+1);
printf("\nPress any key to see the picture.....\n");
getch();
installuserdriver("SVGA256", vgamode);
initgraph(&gd,&gm,"d:\\us\\pcx");
vga256_set_mode();
vga256_set_palette();
if ( (data=(char *)malloc(pcxheader.bytesPerLine))==NULL)
{
 printf("Not enough memory!\n");
 exit(1);
}
if (Equalized) equalize(pcxfile);
if ( get_data(pcxfile,datafile) == -1)
        printf("Error in data!\n");
printf("\a");
getch();
fclose(pcxfile);
fclose(datafile);
free(data);
closegraph();

statistics(gray);
if (Equalized) statistics(equalizedGray);
}
```

🗁 Listing 2: feature.c

This subroutine reads in the equalised gray levels of an image, each time at a 10x10 pixels blocks and extracts the statistical texture features of these subimages (blocks).

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#define stepsize 10
#define winsize 100
#define square(x) (x)*(x)

int data[winsize];
int rc[256][stepsize];
double sqrt (double x);

float meanvalu(float hv[])
{
 int a;
 float meanvalue=0.0F;

 for (a=0; a<256; a++)  meanvalue += a*hv[a];
 return meanvalue;
 }

 void sDeviation(float hv[], float meanvalue, FILE *outfile)
 {
 float variance=0.0F, temp;
```

```
int b;

for (b=0; b<256; b++) {
 temp = b-meanvalue;
 variance += temp*temp*hv[b];
}
fprintf (outfile, "%f\n", sqrt(variance) );
}

void entrop(float hv[], FILE *outfile)
{
 int e;
 float entropy=0.0F;

 for (e=0; e<256; e++) {
  if(hv[e]>(float)1e-5) entropy-=hv[e]*(float)log((double)hv[e]);
 }
 fprintf (outfile, "%f\n", entropy);
}

void grayDif(int map[], FILE *outfile)
{
 int i, j, k, temp, npixel=380;
 float hg[256], meanGrayDif=0.0F;   /* hg=histogram of grayDifference */

 for (k=0; k<256; k++) hg[k]=0;
 for (k=0; k<256; k++) {
  for (i=0; i<(stepsize-1); i++) {
   for (j=0; j<stepsize; j++) {
    temp = abs( map[(i+1)*stepsize+j]-map[i*stepsize+j] );
    if (temp == k) hg[k]++;
   }
  }
  hg[k]=hg[k]/npixel;        /* convert to histogram */
 }
 for (k=0; k<256; k++) meanGrayDif+=k*hg[k];
 fprintf (outfile, "%f\n", meanGrayDif);
}

void runPercent(int map[], FILE *outfile)
{
  register int i, j;
  int run=0;
  float runPC=0.0F;

  for (i=0; i<255; i++) {
   for (j=0; j<stepsize; j++) {
    rc[i][j]=0;
   }
  }

  for (j=0; j<stepsize; j++)
  {
   i=0;
   while (i<(stepsize-1))
   {
    if (map[i*stepsize+j]==map[(i+1)*stepsize+j]) run++;
    else { rc[ map[i*stepsize+j] ][run]++; run=0;}
    i++;
   }
   // when i==9 //
   rc[ map[(i-1)*stepsize+j] ][run]++;
  }

  for (i=0; i<255; i++) {
   for (j=0; j<stepsize; j++) {
    runPC+=rc[i][j];
   }
  }
  runPC=runPC/(winsize);
  fprintf (outfile, "%f\n", runPC);
 }

 void get_fractal(int map[], FILE *outfile)
 {
  int v, x, y, j;
```

```
int graylevel, maxi=0, mini=256;
int scale[4]={2,4,5,10}, scaledata[4], scalelength=0;
float logscale[4]={0.5,0.25,0.2,0.1};
double sumx=0.0F, sumy=0.0F, sumxy=0.0F, sumx2=0.0F, fractal=0.0F;

for (v=0; v<4; v++)
{
 scaledata[v]=0;
 scalelength=stepsize*stepsize/scale[v];
 for (x=0; x<scalelength; x+=scale[v])
 {
  for (y=0; y<scale[v]; y++)
  {
   for (j=0; j<scale[v]; j++)
   {
    if (map[y*scalelength+x+j]>maxi) maxi=map[y*scalelength+x+j];
    if (map[y*scalelength+x+j]<mini) mini=map[y*scalelength+x+j];
   }
  }
  /* counting the blocks occupied by the gray levels */
  if (maxi<scale[v]) maxi=0;
  if (maxi>scale[v] && maxi%scale[v]!=0) maxi=maxi/scale[v]+1;
            else maxi=maxi/scale[v];
  if (mini<scale[v]) mini=0;
  if (mini>scale[v] && mini%scale[v]!=0) mini=mini/scale[v]+1;
            else mini=mini/scale[v];
  scaledata[v]+=maxi-mini+1;
  mini=256;
  maxi=0;
 }  /* loop x */
}

for (v=0; v<4; v++)
{
 sumx += log(logscale[v]);
 sumy += log(scaledata[v]);
 sumxy+= log(scaledata[v]) * log(logscale[v]);
 sumx2+= log(logscale[v]) * log(logscale[v]);
}

 fractal = ( (4*sumxy - sumx*sumy) / (4*sumx2 - sumx*sumx) );
 fprintf (outfile, "%f\n", fractal);
}

void hvEdge(int map[], FILE *outfile)
{
 int a, b, temp=0;
 int vEdge=0, vDir=0;
 int hEdge=0, hDir=0;
 int edge=0, direction=0;
 int threshold=10;
 unsigned char nmap[stepsize][stepsize];

 for (a=0; a<stepsize; a++) {
  for (b=0; b<stepsize; b++){
   nmap[a][b]=0;
  }
 }

 /* detecting horizontal edge */

  for (a=0; a<stepsize; a++) {
  for (b=0; b<(stepsize-1); b++){
   if ( (abs)(map[a*stepsize+b+1]-map[a*stepsize+b])>threshold )
    nmap[a][b+1]=1;
   temp = ( map[a*stepsize+b+1]-map[a*stepsize+b] );
   hDir += temp;
  }
 }

  for (a=0; a<stepsize; a++) {
  for (b=0; b<stepsize; b++) {
   if (nmap[a][b]==1) {
    if ( ((nmap[a][b-1]==1) || (nmap[a][b+1]==1)) &&
                ((nmap[a-1][b-1]==1) || (nmap[a-1][b]==1) || (nmap[a-1][b+1]==1)) &&
                ((nmap[a+1][b-1]==1) || (nmap[a+1][b]==1) || (nmap[a+1][b+1]==1)) )
```

```
            hEdge++;
  }
 }
}

/* detecting vertical edge */

for (a=0; a<(stepsize-1); a++) {
 for (b=0; b<stepsize; b++){
  if ( (abs)(map[(a+1)*stepsize+b]-map[a*stepsize+b])>threshold )
   nmap[a+1][b]=1;
  temp = ( map[(a+1)*stepsize+b]-map[a*stepsize+b] );
  vDir += temp;
 }
}

for (a=0; a<stepsize; a++) {
 for (b=0; b<stepsize; b++) {
  if (nmap[a][b]==1) {
   if ( ((nmap[a][b-1]==1) || (nmap[a][b+1]==1)) &&
              ((nmap[a-1][b-1]==1) || (nmap[a-1][b]==1) || (nmap[a-1][b+1]==1)) &&
              ((nmap[a+1][b-1]==1) || (nmap[a+1][b]==1) || (nmap[a+1][b+1]==1)) )
            vEdge++;
  }
 }
}

edge = hEdge + vEdge;
direction = hDir + vDir;
fprintf (outfile, "%d\n%d\n", direction, edge);
}

void roughness(int map[], FILE *outfile)
{
 int a, b, midPt=stepsize/2;
 float distance;
 float tempR, R=0.0F;  /* R=roughness */

 for (a=0; a<stepsize; a++) {
  for (b=0; b<stepsize; b++){
   distance = (float) ((a-midPt)*(a-midPt)+(b-midPt)*(b-midPt));
   if (distance==0.0) tempR=0.0;
   else tempR = (square((float)map[a*stepsize+b] - (float)map[midPt*stepsize+midPt])) / sqrt(distance);
   R += tempR;
  }
 }
 fprintf (outfile, "%f\n", R/(stepsize*stepsize));
}

void pixDif(int map[], FILE *outfile)
{
 int a, b, temp=0, vDiff=0, hDiff=0;

 for (a=0; a<(stepsize-1); a++) {
  for (b=0; b<stepsize; b++){
   temp = abs(map[(a+1)*stepsize+b]-map[a*stepsize+b]);
   vDiff += temp;
  }
 }

 for (a=0; a<stepsize; a++) {
  for (b=0; b<(stepsize-1); b++){
   temp = abs(map[a*stepsize+b+1]-map[a*stepsize+b]);
   hDiff += temp;
  }
 }

 fprintf (outfile, "%d\n", (hDiff+vDiff));
}

void main()
{
 register int i, k, n, m;
 FILE *infile, *outfile;
 char inname[40], outname[40];
 unsigned char **map;
```

```
int imageHeight, imageWidth, height, width;
int value;
float mean, gray[256], hv[256];
/* hv[]: histogram of grayscale */

printf("Name of file to be processed: ");
gets(inname);
if ((infile = fopen( inname, "rt" ))==NULL) {
   printf("Open %s error!\n", inname);
   exit(1);
}
printf("Name of file to be saved: ");
gets(outname);
if ((outfile = fopen (outname, "w"))==NULL) {
            printf("Open %s error!\n", outname);
            exit(2);
}

fscanf (infile, "%d%d", &imageWidth, &imageHeight);
height = imageHeight/stepsize;
width  = imageWidth/stepsize;

map = (unsigned char **) malloc (sizeof(char*)*stepsize);
for (i=0; i<stepsize; i++)
{
 map[i] = (unsigned char*) malloc (sizeof(unsigned char)*imageWidth);
 if (map[i]==NULL) printf ("No memory available !!!%c\n", 07);
}

printf ("window processing ...\n");
for (i=0; i<height; i++)
{
 for (m=0; m<stepsize; m++)
 {
  for (k=0; k<imageWidth; k++)
  {
   fscanf(infile, "%d", &value);
   map[m][k] = value;
  }
 }
 for (k=0; k<width; k++)
 {
  for (m=0; m<256; m++) gray[m]=0.0;
  for (m=0; m<stepsize; m++)
  {
   for (n=0; n<stepsize; n++)
   {
    data[m*stepsize+n] = map[m][k*stepsize+n];
    gray[ map[m][k*stepsize+n] ]++;
   }
  }
  printf("Window %d\n", i*width+k);
  for (n=0; n<256; n++) hv[n]=gray[n]/winsize;
  mean=meanvalu(hv);
  fprintf(outfile, "#%d\n", i*width+k);
  fprintf (outfile, "%f\n", mean);
  sDeviation(hv, mean, outfile);
  entrop(hv, outfile);
  grayDif(data, outfile);
  runPercent(data, outfile);
  get_fractal(data, outfile);
  hvEdge(data, outfile);
  roughness(data, outfile);
  pixDif(data, outfile);
 }
}
printf ("mean, deviation, entropy, gray difference, run percentage \n");
printf ("fractal, direction, edge, roughness, pix difference \n");
printf ("\a");
fclose(infile);
fclose(outfile);
free(map);
}
```

🗁 Listing 3: detect.c

The subroutine reads in the statisitcal features of an image and segment it into four clusters (anterior capsule, background, cell, and posterior capsule) by comparing the distances between fixed centroids. It then isolates suspecious tumour sites from the cells or the posterior capsule by first computing the mean entropy and mean gray difference of the capsules. The suspicious tumour cells are isolated by setting thresholds. The full description of the algorithims is described in the thesis text. The whole image is analysed in four sections due to the limited memory available under the MS DOS environment.

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#define height 60
#define width 80
#define size  1200      /* 15 row x 80 column; breaks the image into 4 sections */
#define square(x) (x)*(x)

int *imageMap, *tempMap;
float *entropyMap, *diaEntropy, *grayMap, *diaGray;


struct WINDOWS
{
 int winNum, winGroup;
 float wMean, wDeviation, wEntropy, wGray, wRun;
 float wDirect, wEdge, wPixDif, wRough;
};
struct WINDOWS window[size];

int mode_filter(int r, int c, int t[])
{
 int fr, fc, mode, count[2]={0,0};

 for (fr=-1; fr<=1; fr++)
  for (fc=-1; fc<=1; fc++)
   if (fr*fc==0) count[ t[(r+fr)*width+(c+fc)] ]++ ;
   /* 4 connectivity neighbours */
 if (count[0]>count[1]) mode=0;
 else mode=1;

 return mode;
}

int border(int r, int c, int t[])
{
 int fr, fc, tag=1;                    /* tag=1 for tumour; tag=3 for border */

 for (fr=-1; fr<=1; fr++) {
  for (fc=-1; fc<=1; fc++) {
   if (fr*fc!=0) continue;             /* 4 connectivity neighbours */
   if ( t[(r+fr)*width+(c+fc)]==3 ) { tag=3; break; }
  }
 }
 return tag;
}

float sDeviation(int count, float mean, float map[])
{
 int c;
 float deviateSum=0.0F, deviate;

 for (c=0; c<count; c++) deviateSum += square( map[c] - mean );
 deviate = sqrt ( deviateSum/count );
 return deviate;
}

void main()
{
 FILE *outfile, *infile, *supfile;
 char inname[40], outname[40], supname[40];
```

```
char windowNum[5];
register int i, j, k;
int sup, rawCluster[4], tCount=0, dCount=0, position=0, minimum=0;
float mean, deviation, entropy, graydif, run;
float fractal, direction, edge, roughness, pixdif;
float dEntropy=0.0F, dEntropySum=0.0F, eDeviate=0.0F;
float dGray=0.0F, dGraySum=0.0F, gDeviate=0.0F;
float tDistance[3], low_threshold=0.0F, high_threshold=0.0F;
float cdPixDif[2]={2018.2179, 551.8500};
float cdRough[2]={320.9736, 57.6315};
float cdGray[2]={3.0989, 0.8609};
float bcPixDif[2]={ 3353.1503, 2896.5083};
float bcdGrayEnt[2][3]={ {1.1058,3.0989,0.8609}, {0.9032,2.0519,2.4026} };

for (i=0; i<4; i++) rawCluster[i]=0;

imageMap = (int *) malloc (sizeof(int*)*4800);
if (imageMap==NULL) { printf ("Not enough memory for imageMap"); exit(0); }
tempMap = (int *) malloc (sizeof(int*)*4800);
if (tempMap==NULL) { printf ("Not enough memory for tempMap"); exit(0); }
entropyMap = (float *) malloc (sizeof(float*)*4800);
if (entropyMap==NULL) { printf ("Not enough memory for entropyMap"); exit(0); }
grayMap = (float *) malloc (sizeof(float*)*4800);
if (grayMap==NULL) { printf ("Not enough memory for grayMap"); exit(0); }
diaEntropy = (float *) malloc (sizeof(float*)*1000);
if (diaEntropy==NULL) { printf ("Not enough memory for diaEntropy"); exit(0); }
diaGray = (float *) malloc (sizeof(float*)*1000);
if (diaGray==NULL) {printf ("Not enough memory"); exit(0); }

printf ("Name of data file for processing: ");
gets(inname);
if ((infile=fopen(inname, "rt"))==NULL)
{
 printf ("Cannot open %s", inname);
 exit(0);
}
printf ("Name of output file: ");
gets(outname);
if ((outfile=fopen(outname, "wt"))==NULL)
{
 printf ("Cannot open %s", outname);
 exit(0);
}
printf ("Name of supporting tissue (.sup) file: ");
gets(supname);
if ((supfile=fopen(supname, "rt"))==NULL)
{
 printf ("Cannot open %s", supname);
 exit(0);
}

fprintf (outfile, "%s\n", inname);
fprintf (outfile, "%s\n", outname);

for (k=0; k<4; k++)
{
 for (i=0; i<size; i++)
 {
  fscanf(infile, "%s%f%f%f%f%f%f%f%f%f%f", &windowNum, &mean, &deviation,
   &entropy, &graydif, &run, &fractal, &direction, &edge, &roughness, &pixdif);
  window[i].wMean=mean;
  window[i].wDeviation=deviation;
  window[i].wEntropy=entropy;
  window[i].wDirect=direction;
  window[i].wEdge=edge;
  window[i].wGray=graydif;
  window[i].wRun=run;
  window[i].wRough=roughness;
  window[i].wPixDif=pixdif;
  printf (".");
  entropyMap[position+i]=entropy;
  grayMap[position+i]=graydif;
 }

 for (i=0; i<size; i++)
 {
```

```
if (window[i].wMean>=180)
{
 if (k==0) window[i].winGroup=0;
 else window[i].winGroup=3;
}
else if (window[i].wMean>100 && window[i].wMean<180) window[i].winGroup=2;
else window[i].winGroup=1;
}

if (k>0)
{
 for (i=0; i<size; i++)
 {
  if (window[i].winGroup==3)
  {
   tDistance[0] = sqrt ( square(window[i].wPixDif - cdPixDif[0]) );
   tDistance[1] = sqrt ( square(window[i].wPixDif - cdPixDif[1]) );
     if (tDistance[0]<tDistance[1]) window[i].winGroup=2;
   tDistance[0] = sqrt ( square(window[i].wRough - cdRough[0]) );
   tDistance[1] = sqrt ( square(window[i].wRough - cdRough[1]) );
     if (tDistance[0]<tDistance[1]) window[i].winGroup=2;
   tDistance[0] = sqrt ( square(window[i].wGray - cdGray[0]) );
   tDistance[1] = sqrt ( square(window[i].wGray - cdGray[1]) );
     if (tDistance[0]<tDistance[1]) window[i].winGroup=2;
  }
  if (window[i].winGroup==2)  /* "2" is over classifed */
  {
   tDistance[0] = sqrt ( square(window[i].wGray - bcdGrayEnt[0][0]) +  square(window[i].wEntropy-bcdGrayEnt[1][0]) );
   tDistance[1] = sqrt ( square(window[i].wGray - bcdGrayEnt[0][1]) +  square(window[i].wEntropy-bcdGrayEnt[1][1]) );
   tDistance[2] = sqrt ( square(window[i].wGray - bcdGrayEnt[0][2]) +  square(window[i].wEntropy-bcdGrayEnt[1][2]) );
   if (tDistance[1]<tDistance[0]) minimum=1;
   if (tDistance[2]<tDistance[minimum]) minimum=2;
   window[i].winGroup=minimum;
  }
  if (window[i].winGroup==1)  /* "1" is over classifed */
  {
   tDistance[0] = sqrt ( square(window[i].wPixDif - bcPixDif[0]) );
   tDistance[1] = sqrt ( square(window[i].wPixDif - bcPixDif[1]) );
      if (tDistance[0]>tDistance[1]) window[i].winGroup=2;
  }
 }
} /* if k>0 */

 for (i=0; i<(size/width); i++)
 {
  for (j=0; j<width; j++)
  {
   rawCluster[ window[i*width+j].winGroup ]++;
   imageMap[position+(i*width+j)]=window[i*width+j].winGroup;
   fprintf (outfile, "%d", imageMap[position+(i*width+j)]);
  }
  fprintf (outfile, "\n");
  }

  position=position+size;

} /* for k loop */
for (i=0; i<4; i++) fprintf (outfile, "%d->%4d\t", i, rawCluster[i]);
fprintf (outfile, "\n");

for (i=0; i<4800; i++)
{
 fscanf (supfile, "%d", &sup);
 if (sup==1) {
  tempMap[i]=3;
  if (i>size) {
   dEntropySum += entropyMap[i];
   diaEntropy[dCount] = entropyMap[i];
   dGraySum += grayMap[i];
   diaGray[dCount] = grayMap[i];
   dCount++;
  }
 }
 else tempMap[i]=0;   /* tempMap={0,3} */
}
dEntropy = dEntropySum/dCount;
```

```
eDeviate = sDeviation(dCount, dEntropy, diaEntropy);
dGray    = dGraySum/dCount;
gDeviate = sDeviation(dCount, dGray, diaGray);
high_threshold = dGray+3.0*gDeviate;
low_threshold  = dEntropy-0.25*eDeviate;
printf ("dCount=%d\n", dCount);
if (dCount>1000) {
 printf ("\nDiaphragm Count fixed to 1,000!\n\a");
 dCount=1000;
}

for (i=0; i<4800; i++)
{
 if (imageMap[i]==2 && entropyMap[i]>low_threshold) {
  imageMap[i]=4;
  tempMap[i]=1;
  tCount++;
 }
 /* some 4's are over classified from 2 */
 if (imageMap[i]==4 && grayMap[i]>high_threshold) {
  imageMap[i]=2;
  tempMap[i]=0;
  tCount--;
 }
 if (imageMap[i]==3 && entropyMap[i]>dEntropy) {
  imageMap[i]=4;
  tempMap[i]=1;
  tCount++;
 }
}            /* tempMap={0,1,3} */

/* eliminates false signals below posterior capsule */
 for (i=40; i<58; i++) {
  for (j=0; j<width; j++) {
   if ( (tempMap[i*width+j]==3) && (tempMap[(i+1)*width+j]!=1) ) {
    for (k=i+1; k<59; k++) imageMap[k*width+j]=0;
   }
  }
 }

  /* eliminates misclassifying borders to be tumour */
 for (i=1; i<(height-1); i++) {
  for (j=1; j<(width-1); j++) {
   if (tempMap[i*width+j]==1) tempMap[i*width+j]=border(i,j,tempMap);
  }
 }  /* forward elimination */
 for (i=(height-1); i>0; i--) {
  for (j=(width-1); j>0; j--) {
   if (tempMap[i*width+j]==1) tempMap[i*width+j]=border(i,j,tempMap);
  }
 }  /* backward elimination */

 printf (" dEntropy=%f\tCount=%d\tDeviation=%f\n", dEntropy, dCount, eDeviate);
 printf ("    dGray=%f\tCount=%d\tDeviation=%f\n", dGray, dCount, gDeviate);
 fprintf (outfile, "Before suptisu,\t<tumour>: %d\n", tCount);
 tCount=0;

 for (i=0; i<4800; i++) {
  if (tempMap[i]==1) { imageMap[i]=1; tCount++; }
  else imageMap[i]=0;
 }
 fprintf (outfile, "After suptisu,\t<tumour>: %d\n", tCount);
 tCount=0;

 /* eliminate noises */
 for (i=1; i<(height-1); i++) {
  for (j=1; j<(width-1); j++) {
   imageMap[i*width+j]=mode_filter(i,j,imageMap);
   if (imageMap[i*width+j]==1) tCount++;
  }
 }
 fprintf (outfile, "After filter,\t<tumour>: %d\n", tCount);
 tCount=0;

 for (i=0; i<width; i++) imageMap[i]=0;
 for (i=(height-1)*width; i<4800; i++) imageMap[i]=0;
```

```
for (i=0; i<height; i++) {
 for (j=0; j<width; j++) {
  if (j==0 || j==(width-1)) imageMap[i*width+j]=0;
  /* fprintf (outfile, "%d", imageMap[i*width+j]); */
  if (imageMap[i*width+j]==1) {
   tCount++;
   fprintf (outfile, "x");
   }
  else fprintf (outfile, ".");
  }
 fprintf (outfile, "\n");
 }

fprintf (outfile, "\n");
fprintf (outfile, "\t<tumour> %d\n", tCount);

fclose (infile);
fclose (outfile);
fclose (supfile);
free (imageMap);
free (tempMap);
free (entropyMap);
free (grayMap);

}
```

🗁 Listing 4: centroid.c

The subroutines computes the centroid of the set of statistical parameters. To fasten the computation, the average values of the parameters from the training set are used as the initial centroid. For proper computation, the first set of values read from the data file should be used as the initial centroids. In the present setting, the training set is known and the centroid is expected to be close to the mean values of the statistical parameters. Fuzzy logic is used for calculation.

```
/* find the centroid points for clusters by fuzzy logic */
/* Bezdek JC. Pattern recognition with fuzzy objective function algorithms */
/* Plenum Press, New York, 1981 */
/* for normal liver images, 4 clusters: anterior capsule background, cell, posterior capsule */

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#define number 1200                /* number of data in training set */
#define difference 0.01
#define cluster 4
#define TRUE   1
#define FALSE  0

typedef struct
{
 float value, member, memFunc[cluster];
} WIN;

void fuzMember(float y, float v[], float ny[])
{
 int a, b;
 float temp[cluster], temp_ny[cluster];

 for (a=0; a<cluster; a++) {
  temp[a] = fabs(y-v[a]);
  temp_ny[a] = 0.0F;
  }
```

```
for (a=0; a<cluster; a++) {
 for (b=0; b<cluster; b++) {
  if (temp[a]==0) temp[a]=1.000000;
  if (temp[b]==0) temp[b]= 1.000000;
  temp_ny[a] += pow((temp[a]/temp[b]),2);
 }
ny[a] = 1.000000/temp_ny[a];
 }
}

int check(float a[], float b[])
{
 int i, running=TRUE;
 int memDif=0;
 for (i=0; i<cluster; i++) {
  if (( fabs(a[i] - b[i]) ) <= difference )  memDif ++;
 }
 if (memDif == cluster) {
  running = FALSE;
 }
 return running;
}

 void main()
 {
  register int i, j, k;
  FILE *infile, *outfile;
  char inname[40], outname[40], class;
  int running=TRUE, count;
  float mean, deviation, entropy, grayDif, run, fractal;
  float direction, edge, roughness, pixdif;
  float feature[10];
  float *parameter[10];
  float mean_C[cluster]={188.94, 67.55, 117.31, 241.62};
  float deviation_C[cluster]={42.7444, 10.3387, 24.3759, 12.4035};
  float entropy_C[cluster]={3.31, 0.69, 2.03, 2.79};
  float graydif_C[cluster]={12.8661, 3.886, 11.7513, 3.091};
  float run_C[cluster]={.8534, .1919, .6061, .7005};
  float fractal_C[cluster]={2.4565, 2.0678, 2.2712, 2.2597};
  float direction_C[cluster]={948.4416, -5.8036, 166.1011, 286.6923};
  float edge_C[cluster]={178.69, 59.13, 213.61, 18.29};
  float pixdif_C[cluster]={7438.99, 2656.38, 7866.10, 1771.58};
  float roughness_C[cluster]={570.92, 49.59, 153.29, 42.34};
  float nvx[cluster]={0.0F};
  float tempN=0.0F, tempD=0.0F;
  WIN *window;

  if ( (window=(WIN*)malloc(sizeof(WIN)*1200))==NULL )
   printf ("Not enought memory for the windows.\n");

  parameter[0]=mean_C;
  parameter[1]=deviation_C;
  parameter[2]=entropy_C;
  parameter[3]=graydif_C;
  parameter[4]=run_C;
  parameter[5]=fractal_C;
  parameter[6]=direction_C;
  parameter[7]=edge_C;
  parameter[8]=pixdif_C;
  parameter[9]=roughness_C;

  printf("Name of file to be processed: ");
  gets(inname);
  if ((infile = fopen( inname, "rt" ))==NULL) {
            printf("Open %s error!\n", inname);
            exit(1);
  }
  printf ("Name of output file: ");
  gets(outname);
  if ((outfile=fopen(outname, "wt"))==NULL)
  {
   printf ("Cannot open %s", outname);
   exit(0);
  }

  for (k=0; k<10; k++)
```

```
{
 infile=fopen(inname, "rt");
 for (i=0; i<number; i++)
 {
 fscanf(infile, "%s%f%f%f%f%f%f%f%f%f%f", &class,
            &feature[0], &feature[1], &feature[2], &feature[3], &feature[4],
            &feature[5], &feature[6], &feature[7], &feature[8], &feature[9]);
window[i].value=feature[k];
 }
 fclose (infile);

 /* initialise the window membership function */
 for (i=0; i<number; i++)
 {
 fuzMember(window[i].value, parameter[k], window[i].memFunc);
 }

 count=0;
 do {
  for (j=0; j<cluster; j++)
  {
   for (i=0; i<number; i++)
   {
            tempD += pow(window[i].memFunc[j],2);
            tempN += ( pow(window[i].memFunc[j],2) )*window[i].value;
   }
   nvx[j] = tempN/tempD;
   tempN=0.0F;
   tempD=0.0F;
  }
  running=check(parameter[k], nvx);
  for (j=0; j<cluster; j++) parameter[k][j]=nvx[j];
  fuzMember(window[i].value, parameter[k], window[i].memFunc);
  count++;
  printf("%d\n", count);
 } while (running);

 outfile=fopen(outname, "at"); /* required to write the outfile */
 for (i=0; i<cluster; i++)
 {
  printf ("%f\t", nvx[i]);
  fprintf (outfile, "%f\t", nvx[i]);
 }
 fprintf(outfile, "\n");
 }  /* end of loop for k */

 printf ("\a");
 fclose(outfile);
 free(window);
 }
```

Appendix B: Further Readings

I. Textbooks on Computer Vision or Images

1. Ballard DH, Brown CM. Computer Vision, New Jersey, Prentice Hall, 1982.
2. Fairhurst MC. Computer Vision for Robotic Systems: An Introduction, New York, Prentice Hall. 1988.
3. Gonzalez RC, Woods RE. Digital Image Processing, Reading, Addison-Wesley, 1992.
4. Lewis R. Practical Digital Image Processing, New York, Ellis Horwood, 1990.
5. Low A. Introductory Computer Vision and Image Processing. London, McGraw-Hill, 1991.
6. Tomita F, Tsuji S. Computer Analysis of Visual Textures. Boston, Kluwer Academic Publishers, 1990.
7. Vernon D. Machine Vision: Automated Visual Inspection and Robot Vision, New York, Prentice Hall, 1991.

II. Reference Books on Processing Algorithms in C Language

1. Lindley CA. Practical Image Processing in C, New York, John Wiley & Sons, 1990.
2. Parker JR. Practical Computer Vision Using C, New York, John Wiley & Sons, 1995.
3. Pitas I. Digital Image Processing Algorithms, New York, Prentice Hall, 1992

REFERENCES

[Akiyama90]
Akiyama I, Saito T, Nakamura M, Taniguchi N, Itoh K. Tissure characterization by using fractal dimension of B-scan image. IEEE Ultrasonics Symposium, Vol. 3, pp.1353-5, 1990.

[Albregtsen92]
Albregtsen F, Nielsen B, Yogesan K. Fractal dimension, only a fraction of the truth? Proceedings 11th IAPR International Conference on Pattern Recognition, Vol III, p.733-6, 1992.

[Ball67]
Ball GH, Hall DJ. ISODATA, an iterative method of multivariate analysis and pattern classification. Behavioral Science, Vol. 12, pp. 15-155, 1967.

[Ballard82]
Ballard DH, Brown CM. Computer Vision. New Jersey, Prenctice Hall, 1982.

[Beasley81]
Beasley RP, Lin CC, Hwang LY, et al. Hepatocellular carcinoma and Hepatitis B virus. A prospective study of 22,707 men in Taiwan. Lancet, Vol 2, pp. 1129-53, 1981.

[Bellman87]
Bellman RE, Zadeh LA. Decision-making in a fuzzy environment. In: Yager RR, Orchinnikor S, Tong RM, Nguyen HT (eds), Fuzzy sets and applications: selected paper by L. A. Zadeh. New York: John Wiley & Sons, pp. 53-79, 1987.

[Bezdek81]
Bezdek JC. Partitions and relations. In: Pattern Recognition with Fuzzy Objective Function Algorithms, Chapter 2. Bezdek (ed), New York, Plenum Press, 1981.

[Bostros88]
Botros N. Instrumentation and algorithm for ultrasound tissue differentiation. IEEE Engineering in Medicine and Biology Society, 10th Annual International Conference, Vol 3, pp. 1252-3, 1988.

[Botros92]
Botros N. A PC-based system for soft tissue classification. Proceedings, 5th Annual IEEE Symposium on Computer-Based Medical systems, pp. 92-9, 1992.

[Bow92]
Bow ST. Pattern recognition and image preprocessing. New York, Marcel Dekker, 1992.

[Carrasco86]
Carrasco CR, Leymaster CE. Simple tailored microcomputer system for the obstetrical ultrasound laboratory. J Ultrasound Med, Vol 5, pp. 21-23, 1986.

[Chan93]
Chan KL. Ultrasonic tissue characterization using fractal feature. Internation Conference on Acoustic Sensing and Imaging, pp 183-8, March 1993.

[Chang85]
Chang WK, Yeoh EK. Hepatitis B infection in Hong Kong: a serological study of a Chinese population. J Hong Kong Med Assoc, Vol 1, pp. 27-30, 1985

[Chaudhuri92]
Chaudhuri BB, Sarkar N. An efficient approach to compute fractal dimension in texture image. Proceedings. 11th IAPR International Conference on Pattern Recognition. Vol 1, Conference A: Computer Vision and Applications pp. 358-61, 1992.

[Chen79]
Chen PC, Pavlidis T. Segmentation by texture using a cooccurrence matrix and a split and merge algorithm. Computer Graphics and Image Processing, Vol. 10, pp. 172-182, 1979.

[Chen89]
Chen CC, Daponte JS, Fox MD. Fractal feature analysis and classification in medical imaging. IEEE Trans Medical Imaging, Vol 8. No 2, pp. 133-142, 1989.

[ChenLin89]
Chen SY, Lin WC, Chen CT. An expert system for medical image segmentation. SPIE Vol 1092, Medical Imaging III: Imaging Processing, 1989.

[Coleman79]
Coleman GB, Andrews HC. Image segmentation by clustering. Proceedings of the IEEE, Vol 67, No 5, May 1979.

[Conners80]
Conners RW, Harlow CA. A theoretical comparison of texture algorithms. IEEE Trans Patt Anal Machine Intell, vol PAMI 2, pp 204-222. May 1980

[Conrath89]
Conrath BC, Daft CMW, O'Brien Jr WD. Applications of neural network to ultrasound tomography. Proceedings IEEE 1989 Ultrasonics Symposium, Vol 2, pp. 1007-10, 1989.

[Cottone83]
Cottone M. Pia Marceno M, Maringhini A, et al. Ultrasound in the diagnosis of hepatocellular carcinoma associated with cirrhosis. Radiology, Vol 13, pp. 517-519. 1983.

[DaPonte88]
DaPonte JS, Gelber J, Fox MD. Effect of cooccurrence displacement vector on quantitating ultrasonic image texture. Proceedings of the Fourteenth (IEEE) Annual Northeast Bioengineering Conference, pp. 298-300, 1988.

[DaPonte89]
DaPonte JS, Gelber J, Fox MD. Statistical classification of ultrasonic image texture. Proc Fifteen IEEE Annual Northeast Bioengineering Conference, pp. 121-2, 1989.

[Debongnie81]
Debongnie JC, Pauls C, Fievez M, et al. Prospective evaluation of the diagnostic accuracy of liver ultrasonography. Gut, Vol 22, pp. 130-135, 1981.

[Dunn74]
Dunn JC. Some recent investigations of a new fuzzy partitioning algorithm and its application to pattern classification problems. J Cybernetics, Vol 4, No 2, pp. 1-15, 1974.

[Fortin92a]
Fortin CS, Ohley WJ, Gewirtz H. Fractal dimension as a method of segmenting cardiac images. IEEE Engineering in Medicine and Biology Magazine, Vol II, Iss 2, pp. 65-71, June 1992.

[Fortin92b]
Fortin CS, Ohley WJ, Gewirtz H. Probability density in the analysis of medical images. Proceedings of the 18th IEEE Annual Northeast Bioengineering Conference pp.141-2, 1992.

[Fukada80]
Fukada Y. Spatial clustering procedures for region analysis. Pattern Recognition, Vol 12, No. 6, pp. 395-403, 1980.

[Galloway75]
Galloway MM. Texture analysis using gray level run lengths. Computer vision and image processing, Vol. 4, pp. 172-179, 1975.

[Gosink79]
Gosink BB, Lemon SK, Scheible W, et al. Accuracy of ultrasonography in diagnosis of hepatocellular disease. Am J Radiology, Vol 133, pp. 19-23, 1979.

[Green77]
Green B, Bree RL, Goldstein HM, Stanley C. Gray scale ultrasound evaluation of hepatic neoplasms: patterns and correlations. Radiology Vol 124, pp. 203-208, 1977.

[Haralick73]
Haralick RM, Dinstein I. Texture features for image classfication. IEEE Trans Syst, Man, Cybern, Vol SMC-3, pp. 610-621, Nov 1973.

[Haralick85]
Haralick RM, Shapiro LG. SURVEY: Image segmentation techniques,. Comput. Vision, Graphics, and Image Process. Vol 29, pp. 100-132, 1985.

[Haralick92]
Haralick RM, Shapiro LG. Computer and Robot Vision, Vol 1, Reading, Addison-Wesley 1992.

[Hecht-Nielsen89]
Hecht-Nielsen R. Neurocomputing. Reading, Addison-Wesley, 1989.

[HongKong94]
Hong Kong Government, Department of Health. Annual Departmental Report 1994-95. Department of Health, Hong Kong Government 1994.

[Hu90]
Hu ZP, Pun T, Pellegrini C. An expert system for guiding image segmentation. Computerized Medical Imaging and Graphics, Vol 4, pp. 13-24, 1990.

[Hussey85]
Hussey M. Basic physics and technology of medical diagnostic ultrasound. London: MacMillan, p. 25, 1985.

[Insana87]
Insana MF, Wagner RF, Gara BS, Momenan R, Shawker TH. Pattern recognition methods for optimizing multivariate tissue signature in diagnostic ultrasound. Ultrasound imaging, Vol 8, pp. 165-180, 1986.

[Itoh85]
Itoh K, Yasuda Y, Aihara T, Koyano A, Konishi T. Acoustic intensity histogram pattern diagnosis of liver diseases. J Clin Ultrasound Vol 13, pp. 499-456, 1985.

[Jang90]
Jang J, Rajala SA. Segmentation based image coding using fractals and the human visual system. 1990 International Conference on Acoustics, Speech and Signal Processing, Vol 4, pp.1957-60, April 1990.

[Julesz73]
Julesz B, et al. Inability of humans to discriminate between visual textures that agree in second-order statistics — Revised. Perception, Vol 2, pp. 391-405, 1973.

[Julesz75]
Julesz B. Experiements in visual properties of texture. Scientific America, Vol 232, pp. 34-43, 1975.

[Kang92]
Kang JY, Lee TP, Yap I, Lun KC. Analysis of cost-effectiveness of different strategies for hepatocellular carcinoma screening in hepatitis B virus carriers. J Gastroenterolo Hepatol, Vol. 7, pp. 463-8, 1992.

[Keller89]
Keller JM, Chen S. Texture description and segmentation through fractal geometry. Computer Vision, Graphics, and Image Processing Vol 45, pp. 150-166, 1989.

[Kim91]
Kim SI, Choi KC, Lee DS. Texture classification using run difference matrix. IEEE Ultrasonics Symposium, Vol. 2, pp. 1097-1100, 1991.

[King85]
King DL, Lizzi FL, Feleppa EJ, et al. Focal and diffuse liver disease studied by quantitative microstructural sonography. Radiology, Vol 155, pp. 457-462, 1985.

[Kremkau86]
Kremkau FW, Taylor KJW. Artifacts in ultrasound imaging. J Ultrasound Med Vol5, pp. 227-37, 1986.

[Layer90]
Layer G, Zuna I, Lorenz A, Zerban H, Haberkorn U, Bannasch P, van Kaick G, Rath U. Computerized ultrasound B-scantexture analysis of experimental fatty liver disease: influence of total lipid content and fat deposit distribution. Ultrasound Imaging, Vol 12, pp. 171-188, 1990.

[Ledley88]
Ledley RS, Ayers WR. Computerized medical imaging and graphics evolves from computerized tomography. Computerized Medical Imaging and Graphics, [editorial] Vol 12, pp. v-xviii, 1988.

[Lerski79]
Lerski AR, Barnett E, Morley P, Mills PR, Watkinson G, MacSween RNM. Computer analysis of ultrasonic signals in diffuse liver disease. Ultrasound Med Biol Vol 5, pp. 341-350, 1979.

[Leung93]
Leung KS, Wong FWS, Chan SWK. Medical image analysis using an expert system shell. In: Hedley AJ, Ho LM, Johnston J, et al (eds). Proceedings of the Second Hong Hong (Asia-Pacific) Medical Informatics Conference. Hong Kong, The Hong Kong Society of Medical Informatics and the Hong Kong Computer society, pp. 83-89, 1993.

[Levine85]
Levine MD. Vision in man and machine. New York, McGrow Hill, 1985.

[Lizzi83]
Lizzi FL, Rorke M, Feleppa EJ, et al. Characterization of liver using families of parameters derived from spectrum analysis. Abstracts, Ultrasonic Imaging and Tissue Characterization Symposium. Ultrasonic imaging, Vol 5, p. 188, 1983.

[Low91]
Low A. Introductory computer vision and image processing. London, McGraw-Hill, p. 2, 1991.

[Mandelbrot82]
Mandelbrot BB. The fractal geometry of nature. New York, Freeman, 1982.

[Mailloux83]
Mailloux GE, Stampfler R, Etheir S, et al. B-scan texture analysis by image segmentation. Abstracts, Ultrasonic Imaging and Tissue Characterization Symposium. Ultrasonic Imaging, Vol 5, p. 190, 1983.

[Mailloux84]
Mailloux GE, Bertrand M, Stampfler R, Etheir S. Texture analysis of ultrasound B-mode images by segmentation. Ultrasonic Imaging, Vol 6, pp. 262-277, 1984.

[McEachon89]
McEachon DL, Hess S, Knecht LB, True LD. Image processing for the rest of us: the potentail utility of inexpensive computerized image analysis in clinical pathology and radiology. Computerized Medical Imaging and Graphics, Vol 13, No 1, pp. 3-30, 1989.

[Medioni84]
Medioni G, Yasumoto Y. a note on using the fractal dimension for segmentation, IEEE Computer Vision Workshop, annapolis, MD, pp. 25-30, 1984.

[Momenan88]
Momenan R, Wagner RF, Loew MH, Insana MF, Garra BS. Characterization of tissue from ultrasound images. IEEE Control Systems Magazine, Vol 8, No 3, pp. 49-53, 1988.

[Momenan89]
Momenan R, Loew MH, Wagner RF, Insana MF, Garra BS. Application of pattern recognition techniques in ultrasound tissue characterization. Proc IEEE 11th Annual International Conference, Vol. 2, pp. 411-2, 1989.

[Mosquera92]
Mosquera A, Cabello D, Carreira MJ, Penedo MG. A fractal-based approach to texture segmentation. International Conference on Image Processing and its Application, pp 450-3 April 1992.

[Muzzolini91]
Muzzolini R, Yang YH, Pierson R. A framework for the evaluation of texture features with application to diagnositc ultrasound images. IEEE Nuclear Science Symposium and Medical Imaging Conference, Sante Fe, NM, Vol 3, pp. 2201-4, 1991.

[Nakashima87]
Nakashima T, Kojiro M. Hepatocellular carcinoma: an atlas of its pathology. Tokyo: Springer-Verlag. 1987.

[Ng93]
Ng V, Coldman A. Diagnosis of melanoma with fractal dimension. Proceedings TENCON '93. 1993 IEEE Region 10 Conference on Computer, Communication, Control and Power Engineering, Vol 4, pp. 514-17, Oct 1993.

[O'Gorman88]
O'Gorman L. A note on histogram equalization for optimal intensity range utilization. Computer Vision, Graphics, and Image Processing, Vol 41, pp. 229-232, 1988.

[Okuda81]
Okuda K. Advances in hepatobiliary ultrasonography. Hepatology, Vol 1, pp. 662-672, 1981.

[Ott86]
Ott WJ. The design and implementation of a computer-based ultrasound data system. J Ultrasound Med Vol 5, pp. 25-32, 1986.

[Pentland84]
Pentland AP. Fractal-based description of natural scenes. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol PAMI-6, No. 6, pp. 661-675, November 1984.

[Raeth85]
Raeth U, Schlaps D, Limberg B, Zuna I, Lorenz A, van Kaick G, Lorenz WJ, Kommerell B. Diagnostic accuracy of computerized B-scan texture analysis and conventional ultrasonography in diffuse parenchymal and malignant liver disease. J Clin Ultrasound, Vol 13, pp.87-99, 1985.

[Ragade77]
Ragade RK, Gupta MM. Fuzzy set theory: introduction. In: Gupta MM, Saridis GN, Gaines BR (eds), Fuzzy automata and decision process. New York: Elservier North-Holland, pp.105-131, 1977.

[Rao90]
Rao AR. Disordered textures. In: A taxonomy for texture description and identification, Chap 5, pp. 145-157, New York, Springer-Verlag, 1990.

[Rhys90]
Rhys L. Practical digital image processing. New York, Ellis Horwood 1990.

[Rosenbusch89]
Rosenbusch G, Smits NJ, Reeders JWAJ. Ultrasonography in hepatobiliary and pancreatic malignancies. In: Lygidakis NJ, Tytgat GNJ (eds). Hepatobiliary and pancreatic malignancies. Diagnosis, medical and surgical management. New York: Thieme Medical Publishers, pp 51-65, 1989.

[Schalkoff92]
Schalkoff R. Pattern recognition: statistical, structural, and neural approaches. New York, John Wiley, 1992.

[Schouten93]
Schouten Th E, klein Gebbinck M, Thijssen JM, Verhoeven JTM. Ultrasonic tissue characterisation using neural networks. Third International Conference on Artificial Nerual Networks, pp. 110-2, 1993.

[Schuster88]
Schuster E, Knoflach P, Grabner G. Local texture analysis: an approach to differentiating liver tissue objectively. J Clin Ultrasound, Vol 16, pp. 453-461, 1988.

[Sheu84]
Sheu JC, Chen DS, Sung JL, et al. Ultrasonography of small hepatic tumours using high-resolution linear-array real-time instruments. Radiology, Vol 150, pp. 797-802, 1984.

[Sheu85a]
Sheu JC, Chen DS, Sung JL, et al. Hepatocellular carcinoma: US evolution in the early stage. Radiology Vol 155, pp. 463-467, 1985.

[Sheu85b]
Sheu JC, Sung JL, Chen DS, et al. Early detection of hepatocellular carcinoma by real-time ultrasonography. Cancer, Vol 56, pp. 660-6, 1985.

[Tanaka83]
Tanaka S, Kitamura T, Imaoka S, et al. Hepatocellular carcinoma: sonographic and histologic correlation. Am J Radiology, Vol 140, pp.701-707, 1983.

[Thomas91]
Thomas J, Peters RA, Jeanty P. Automatic segmentation of ultrasound images using morphological operators. IEEE Trans Medical Imaging, Vol. 10, No. 2, pp. 180-6, 1991.

[Tjahjadi89]
Tjahjadi T, Henson R. A knowledge based system for image understanding. Image Processing and Its Application. IEE Conf Publ, Vol 07, pp. 88-92, 1989.

[Tomita90]
Tomita F, Tsuji S. Computer analysis of visual textures. Boston, Kluwer Academic Publisher, 1990.

[Tou74]
Tou JT, Gonzalez RC. Pattern recognition principles, London, Addison Wesley, 1974.

[Weill87]
Weill FS. Ultrasound diagnosis of digestive disease. 3rd edition. Berlin: Springer-Verlag, 1987.

[Wells89]
Wells PNT, Berry E. Medical image processing. Image Processing and its Application IEE Conf Publication 307, pp. 310-313, 1989.

[Weszka76a]
Weszka JS, Dyer CR, Rosenfeld A. A comparative study of texture measures for terrain classification. IEEE Trans, Vol SMC-6, pp. 269-285, 1976.

[Weszka76b]
Weszka JS, Rosenfeld A. An application of texture analysis to material inspection. Pattern Recognition, Vol 8, pp. 195-200, 1976.

[Wong93]
Wong SH, Chan KL, Fung PW. Automatic segmentation of ultrasonic image. Proceedings TENCON '93. IEEE Region 10 Conference on Computer, Communication, Control and Power Engineering, Vol. 2, pp. 910-13, 1993.

[Youssef89]
Youssef A, Shehada R. Ultrasound textural parameters for hepatic masses evaluation. Proc 11th IEEE Annual International Conference, Vol. 2, pp. 415-6, 1989.

[Zadeh65]
Zadeh L. Fuzzy sets, Inform and Control, Vol 8, pp. 338-353, 1965.

[Zatari93]
Zatari D, Botros N. In-vivo ultrasound liver differentiation using artificial neural network. Proceedings of Sixth Annual IEEE Symposium on Cmputer-Based Medical System, pp. 223-8, 1993.

[Zhang89]
Zhang P, Barad HS, Martinez AB. Applications of fractal modeling to cell image analysis. IEEE Southeastcon 1989 Proceedings. Energy and Information Technologies in the southeast, Vol. 2, pp.618-22, April 1989.

[Zhang92]
Zhang H, Zhao H. Microcomputer BASIC program for quantitative analysis of biological shape. Computer Methods and Programs in Biomedicine Vol 37, pp. 159-161, 1992.